creative computing

Mar-Apr 1977
vol 3, no 2

Austria Sch 32.00
Belgium BF 70.00
Canada $1.50
Denmark Kr 11.00
Finland Mk 7.00
France F 9.00
Germany DM 4.50
Great Britain £ 1.00
Greece Dr. 65.00
Holland Dfl 5.00
Italy L 1500
Norway Kr 10.00
Portugal Esc 55.00
Spain Ptas 120.00
Sweden Kr 8.00
Switzerland SF 4.50
USA $1.50

the magazine of recreational and educational computing

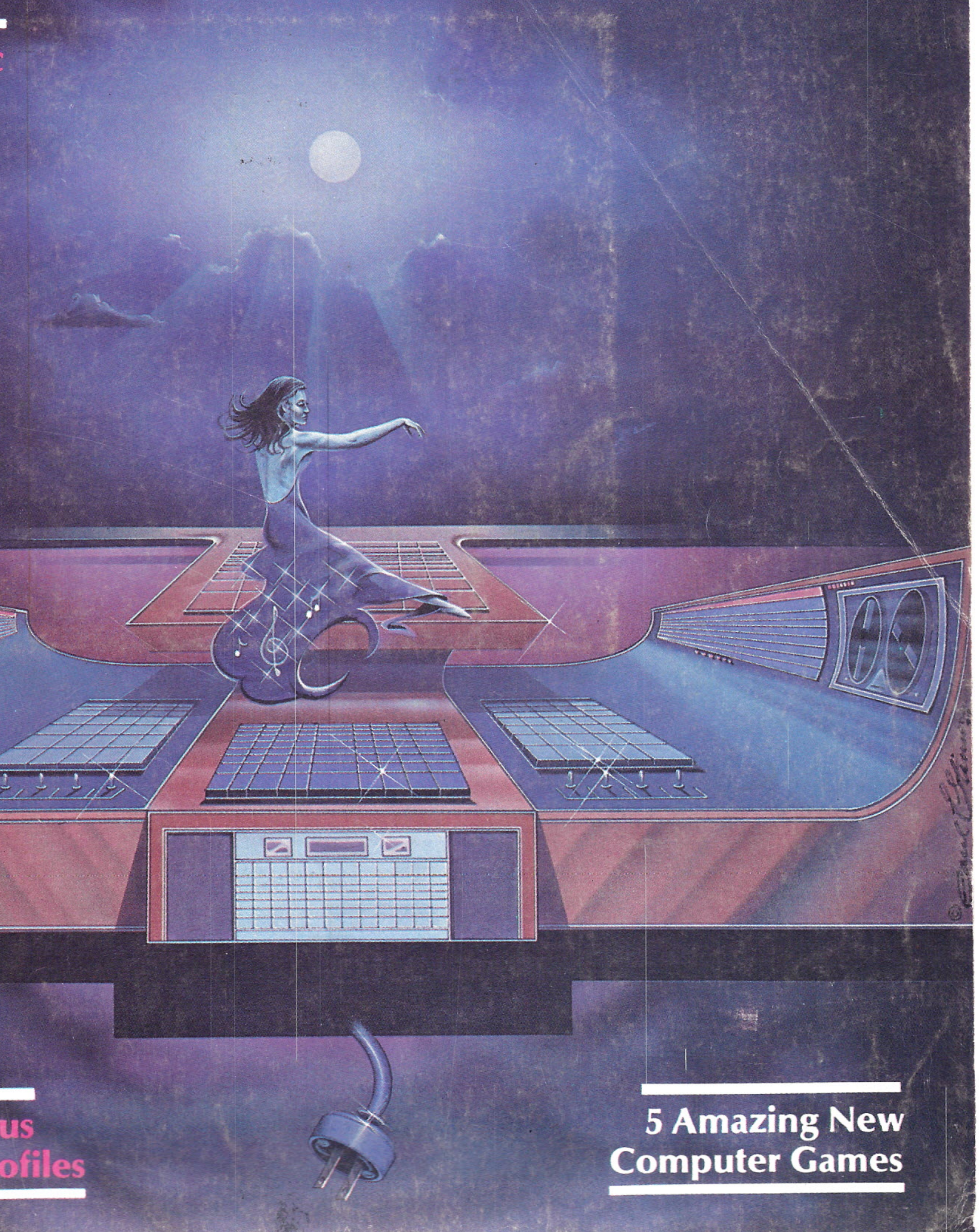
$1.50

Special Music Features

Saga of a System

Algorithmic BASIC

Build a Kluge Harp



Altair S-100 Bus Peripheral Profiles

5 Amazing New Computer Games

SWTPC 6800

- ★ COMPLETE WITH 2K OF MEMORY
- ★ SERIAL INTERFACE
- ★ STANDARD ROM MONITOR (Motorola MC 6830L7)
- ★ 4K AND 8K BASIC AVAILABLE

Always the best value in hardware and now an outstanding selection of software too. What more could you want in a computer system? For less than four hundred dollars you get everything you need—ready to connect to a terminal and go to work. No surprises no funny business, just good reliable hardware in a very practical system that may be expanded to meet almost any later need.

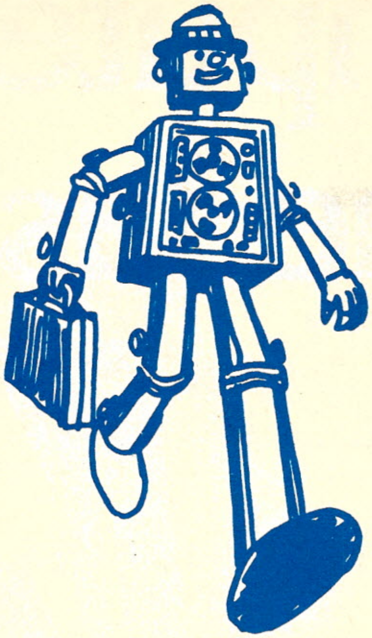
Memory and interfaces are not extra cost items in our system. A standard Motorola MIKBUG® ROM monitor makes the system completely compatible with Motorola software and eliminates any need for console switches and light. Data may be entered from the terminal in convenient hexidecimal form. The power supply is adequate to operate a fully expanded system with up to 24K of memory and up to eight (8) interfaces—simultaneously.

See the 6800 and our peripheral equipment at your nearest dealer, or write for a complete description.

MP-68 COMPUTER KIT—with serial interface, 2k of memory and ROM monitor $395.00 ppd

® Motorola

Southwest Technical Products Corporation, 219 W. Rhapsody, San Antonio, Texas 78216



MEMBER
COSMEP

Under 1,000,000 Circulation

creative computing

Mar-Apr 1977 Volume 3, Number 2 Consecutive Issue No. 14

Creative Computing magazine is published bi-monthly by Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. (Editorial office: 20 Lynnfield Drive, Morristown, NJ 07960)

Domestic Subscriptions: Institutional 1-year $15, 3-year $40; Individual 1 year $8, 3-year $21. Subscription orders, change of address, P.O. Form 3579 to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960
Foreign Subscriptions:
Great Britain. Institutional 1-year £9.40, 3-year £25.00; Individual 1-year £6.25, 3-year £16.90. Orders to Creative Computing, 60 Porchester Road, Southampton SOZ 7 JD, England.
Europe. 1-year subscription. Austria Sch 180, Belguim BF 400, Denmark Kr 65, Finland Mk 40, France Fr 50, Germany DM 25, Greece Dr 360, Holland Df1 27, Italy L 8500, Norway kr 55, Portugal Esc 320, Spain Ptas 700, Sweden kr 45, Switzerland SF 25. Orders to Pan Atlantic Computer Systems gmbh, Frankfurter Strasse 78, D61 Darmstadt, Germany Fed Rep.
Other Countries. 1-year $10, 3-year $27. Orders to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960.

Second class postage paid at Morristown, New Jersey and at additional mailing offices.

in this issue . . .

ARTICLES

FICTION AND FOOLISHNESS

REVIEWS AND RESOURCES

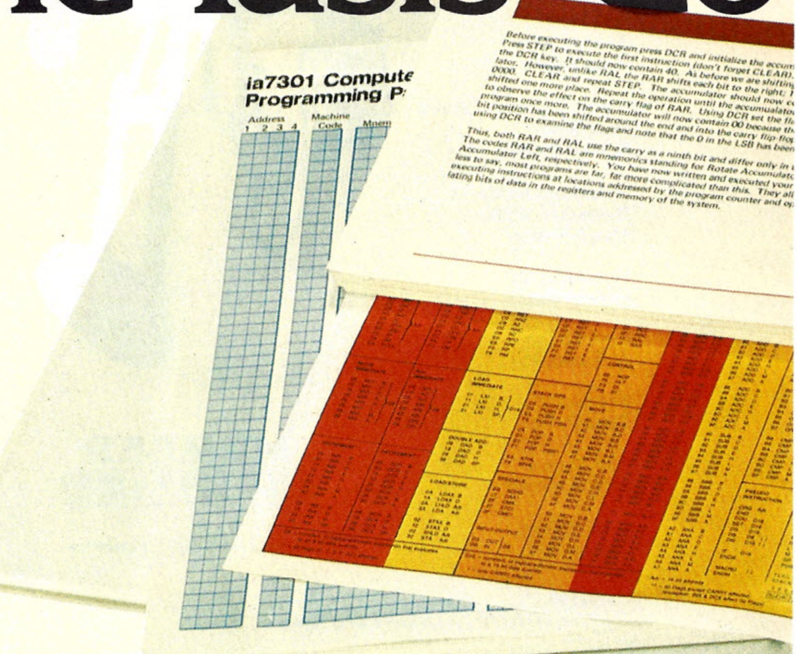
THINGS TO DO - COMPUTER GAMES

DEPARTMENTS

THE COVER

The cover is an original acrylic painting by the incredibly talented Paul Stinson. It was done especially for this special music issue and depicts the symbiosis between music, dance, and the computer. Paul's works may be seen at many galleries as well as at most major science fiction conventions.

Microcomputer pro
with the Iasis Co



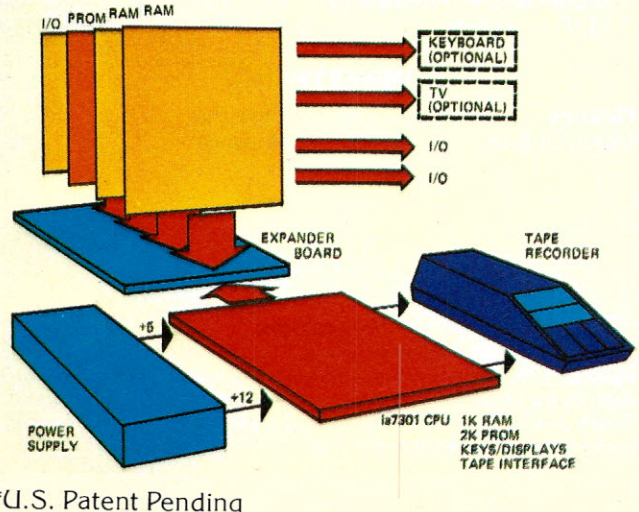
The fact is that right now microcomputer programming is a bear. Microprocessors are loaded with subtleties which make software development a long, arduous process. That's why we developed the ia7301 Computer in a Book.* It's a fully operational microcomputer system and a 250 page programming course all contained in a 3-ring binder. This is not a kit or a toy but a powerful, microcomputer system (based on the industry standard, the 8080) and a practical programming course specifically designed to quickly bring you up to a high level of understanding and proficiency in programming 8080 based microcomputer systems.

The Computer in a Book comes to you completely assembled and tested. All you need is an inexpensive dual voltage (+12V & +5V) power supply. The −5V is generated internally in the computer. There is nothing else to buy.

A super programming course

The programming course text is easy to follow and begins with a one instruction program to determine if a switch is open or closed. This is built upon and expanded, instruction by instruction, until 250 pages later, you become adept at programming complex problems like multi-byte arithmetic and games of skill like Pong.™ Only with the Iasis Computer in a Book can you have the advantages of a handy programming text together with an operational computer to load and test programs each step of the way and thereby learn the intricacies of microcomputer programming at a comfortable pace.

And since this microcomputer has a special built in monitor program which allows you to look into the operational parts of the system you'll never get bogged down in debugging or editing. The ia7301 Computer in a Book is the fastest way to learn everything about microcomputer programming.



*U.S. Patent Pending
Pong is a trademark of Atari, Inc.

Some great microcomputer features, too

The microcomputer system features a 24 pad keyboard, 8 seven segment LED readouts that display information in hexadecimal code which is far more versatile and advanced than binary or octal coded systems, and an onboard cassette tape interface for saving programs. The hexadecimal keyboard also contains 6 special mode keys which allow you to call up and change any data or instructions in the 8080 registers or in the system's RAM memory. Likewise programs can be executed instantly or they can be stepped through one instruction at a time using the appropriate mode key, so that you learn your way around the inner workings of an entire microcomputer system.

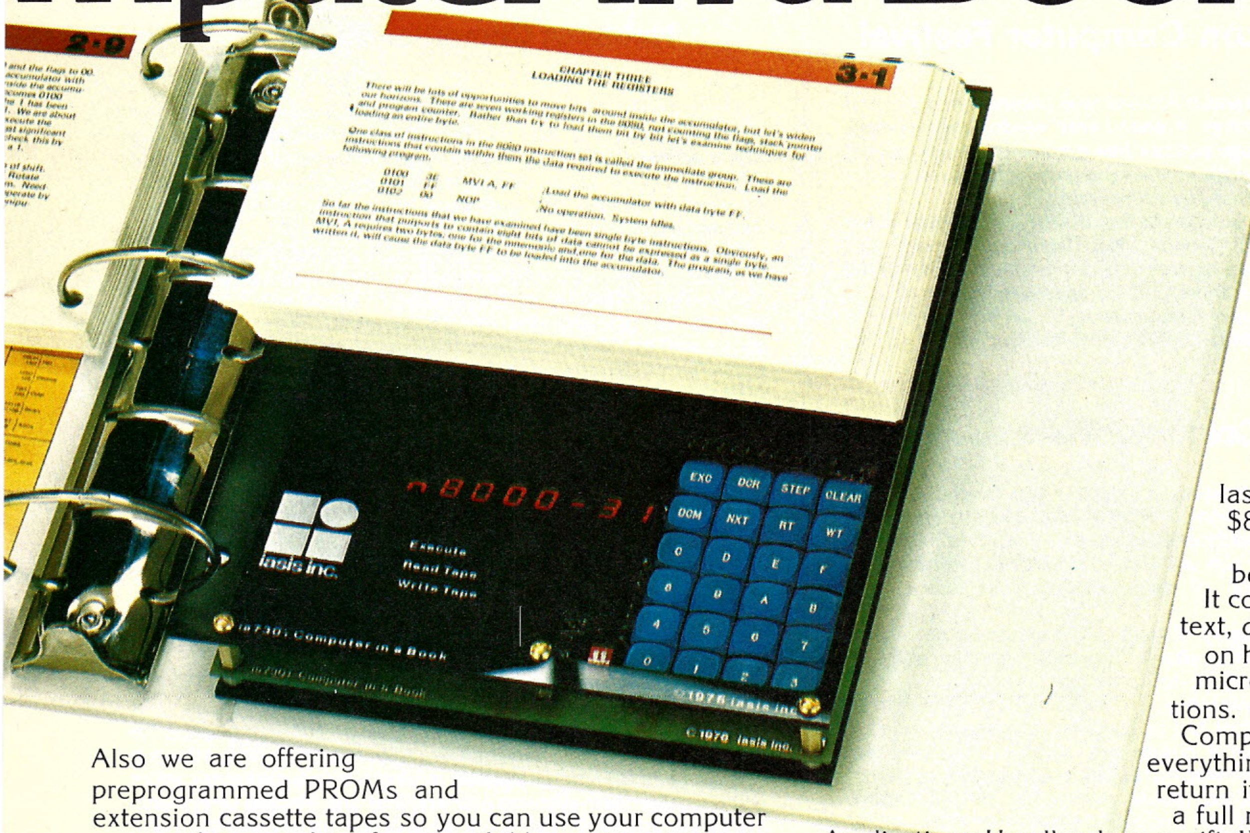
Also the write tape and read tape mode keys have been carefully designed for accurate and convenient operation with any home cassette tape recorder that has an earphone and remote microphone jack. Two LED indicator lamps tell how long it takes to dump or reload programs from the system's memory onto tape and back again. But in the reloading cycle, if any errors have occurred such as a lost piece of data, or the volume knob is too low, the readout displays will indicate errors. This little feature prevents untold problems in debugging a reloaded program.

Upwards expandability from the start

We designed the Computer in a Book to be upwards expandable and not become a kluge in the process. The microcomputer contains 1K bytes of RAM memory, 1K bytes of PROM memory (containing the monitor program), and 2 I/O ports. The Computer in a Book is expandable to virtually any level you want, i.e. up to 65K bytes of memory and 256 I/O ports.

Optional expander boards are available and attach to the ia7301 computer at the top edge connector. A wide variety of standard interface boards can be plugged into the system to give add on memory, TV and teletype interface, and much more. Thus an educational system is easily upgraded into a full computer system.

gramming is a snap
mputer in a Book

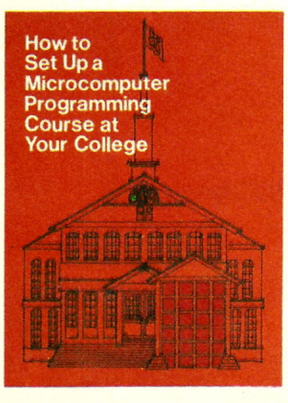


A free bonus

If you order your Computer in a Book before May 15, 1977, Iasis will give you an $8.00 Microcomputer Applications Handbook as a free bonus. It contains 144 pages of text, diagrams, and tables on hardware design and microcomputer applications. Order today. If the Computer in a Book isn't everything we say it is, then return it within 15 days for a full refund and keep the Applications Handbook as a gift. We're sure you'll find that microcomputer programming is a snap with the ia7301 Computer in a Book.

Also we are offering preprogrammed PROMs and extension cassette tapes so you can use your computer as a teaching machine for your children, as an inventory control system for your business or as an accounting system for your home. The Computer in a Book can become a computer for all reasons.

A college assistance program



Educators interested in exposing their students to a comprehensive background in Microcomputer programming should look into the Iasis Microcomputer Instructional Courses for their college or university. Send for our free pamphlet which describes ways of setting up short microcomputer programming courses. It offers some advice on structuring a coordinated and comprehensive program, so your students can learn programming and get valuable hands-on experience with operational systems at very reasonable prices.

The price

The complete Computer in a Book which includes an operational 8080 based system, 250 page programming course, machine code pad, hexadecimal conversion card all in a 3-ring binder is offered for only $450. The Computer in a Book has a 90 day parts and service warranty. Iasis also provides a check out list and start up instructions with each system. Please allow 30 days for delivery.

☐ Here's my check or money order for ________ complete ia7301 Computer(s) in a Book at $450 each. Since I ordered my Computer in a Book before May 15, 1977, I want the Microcomputer Applications Handbook as a free bonus. (Calif. residents add 6½% sales tax).

☐ Charge my order to the credit card below:
BankAmericard No.________________
Master Charge No.________________
For Master Charge card, put 4 digit number from above name here________________
My Signature________________
Credit card expiration date________________

Name________________
Firm________________
Address________________
City__________ State__________ Zip__________

☐ Please send me your pamphlet on setting up a Microcomputer Programming Course.

Mail to: Iasis Inc., 815 W. Maude, Suite 32, Sunnyvale, CA 94086

(408) 732-5700

iasis inc.

2nd Trenton Computer Festival

Last year the Trenton Festival was planned as a modest hobbyist get-together. It ended with vendor booths lining the corridors, a huge outdoor flea market and sessions in 3 buildings. This year the Festival promises to be twice as good because it's *two* days instead of one.

It will be in the Student Center Building at Trenton State College, Route 31, Trenton, NJ. There will be some 85 booths, a huge flea market, and 30 outstanding speakers (including our own Dave Ahl). Co-sponsored by the NJ and Phila. Area Computer Clubs. Starts 10 am April 30-May 1. Registration is a delightfully modest $4.00; students $2.00.

Computer Faire

Have you got your tickets yet for the Faire "happening" April 15-17 in Brook's Hall, San Francisco? You can even get course credit from the University of California by attending. It's shaping up to a magnificent Faire with something for everyone and with the emphasis on use of microcomputers at home and *in education.*

There will be a display of over 200 systems up and running which might contain ones you've laid awake dreaming about, and just needed to see working to figure out how to do.

The over 100 conference sections are planned for new information and a lot of *fun* (a theme in *this* show), to tell you about graphics on home computers, computer assisted music systems, floppy disc systems for personal computers, systems for small business or classes, games, standards. The list is long.

What's impressive about this Faire is the local support that is co-sponsoring it, which includes amateur, professional, and educational groups.

There will be a banquet both Friday and Saturday night, each to feature two speakers, one "real world" and one "future world." Friday night's speakers are John Witney, who will show his new computer animation films, and Frederick Pohl, who will weave a science fiction future.

Featured on Saturday night are "real world" Henry Tropp, who, with help from a joint AFIPS-Smithsonian grant, has interviewed on video tape many of the old timers who developed the technology of computers. He will recount their anecdotes and viewpoints. The "future world" speaker is none other than Ted Nelson whose "Unforgettable Next Two Years" you'll never forget.

Even for all of us who hate banquets these alone sound like they'd make going to the conference worthwhile.

Creative Computing will be at the Faire with some conference-specials for our readers. Meet us at booth 214.

For tickets or details contact: Jim Warren, Faire Chairbody, Box 1579, Palo Alto, CA 94302, (415) 851-7664 or 323-3111.

15th Annual AEDS Convention

Held at the Green Oaks Inn in Fort Worth, Texas during April 25-29, 1977, the theme of this conference is EDS = f (H,S,P): Educational Data Systems are a function of Hardware, Software, and Peopleware.

Papers will focus on new practical, workable approaches to problems in the educational data systems field.

A.U. Majors, Convention Chairman, Ft. Worth ISD, Ft. Worth, Texas.

Hobbyist Computer Shows

Three hobbyist computer shows will be held this spring and summer beginning with the first Western Personal Computing Show at the Los Angeles Airport Hyatt House Hotel, March 19 and 20. The second show will be held at the City Line Marriott in Philadelphia, April 30 and May 1, and the third at Haynes Auditorium in Boston, June 18-19.

In addition to manufacturer exhibits, these shows will feature a number of seminars and workshops. One seminar of note is "Software — The Tip of the Iceberg?" which will cover many aspects of software for micorcomputers. These shows are sponsored by *Personal Computing* Magazine. Registration is $10 at the door or $7.50 in advance.

Personal Computing Shows, Conference and Exposition Management Co., Box 844, Greenwich, CT 06830.

15th Annual Southeastern Regional ACM Conference

Hobbyists are encouraged to attend this conference to exchange ideas, see micro-systems in operation and attend a special technical paper session in which micro-computer users will describe their systems and applications. Papers or systems are being solicited. There will be a computer chess match and large exhibit area. $2 for a pass to exhibits and the computer match; $5 adds the hobbyist technical paper session; $30 adds a table with power outlets to display a system; more $ for more. The conference is being held in Biloxi, Mississippi on the Gulf Coast, April 18-20, 1977.

To present a paper, exhibit, or get information contact: USM-ACM, Robert Hyatt, Exhibit Chairperson, Box 5251, Southern Station, Hattiesburg, MS, 39401, (601) 266-7131.

Survey Drawing Winners

In the Nov/Dec 1976 issue of *Creative* we included a detailed reader questionnaire about you and your interests. To everyone who returned a questionnaire: a big THANK YOU for your thoughtful response.

You'll recall that we also offered a crass, materialistic incentive for returning the questionnaire, namely 3 prizes. The lucky winners are:

- Ed Anglin, Santa Barbara, Calif. (3-year subscription)
- John Rabenaldt, Odessa, Texas (1-year sub'n)
- Jim Denning, San Francisco, Calif. (1-year sub'n)

Questionnaires were numbered as they arrived in our office. On Jan. 12 the winners were picked by computer, of course. Or, to be more exact, by the Basic random number generator in the GE Timesharing System.

A complete analysis of the contents of the questionnaires will appear in the May/June issue.

The Small Computer

Twenty-five years ago a computer as powerful as the new Processor Technology Sol-20 priced out at a cool million.

Now for only $995 in kit form or $1495 fully assembled and tested you can have your own small computer with perhaps even more power. It comes in a package about the size of a typewriter. And there's nothing like it on the market today. Not from IBM, Burroughs, DEC, HP or anybody else!

It fills a new role

If you're an engineer, scientist or businessman, the Sol-20 can help you solve many or all of your design problems, help you quantify research, and handle the books too. For not much more than the price of a good calculator, you can have high level computer power.

Use it in the office, lab, plant or home

Sol-20 is a smart terminal for distributed processing. Sol-20 is a stand alone computer for data collection, handling and analysis. Sol-20 is a text editor. In fact, Sol-20 is the key element of a full fledged computer system including hardware, software and peripheral gear. It's a computer system with a keyboard, extra memory, I/O interfaces, factory backup, service notes, users group.

It's a computer you can take home after hours to play or create sophisticated games, do your personal books and taxes, and a whole host of other tasks.

Those of you who are familiar with small computers will recognize what an advance the Sol-20 is.

Sol-20 offers all these features as standard:

8080 microprocessor — 1024 character video display circuitry — control PROM memory — 1024 words of static low-power RAM — 1024 words of preprogrammed PROM — built-in cassette interface capable of controlling two recorders at 1200 bits per second — both parallel and serial standardized interface connectors — a complete power supply including ultra quiet fan — a beautiful case with solid walnut sides — software which includes a preprogrammed PROM personality module and a data cassette with BASIC-5 language plus two sophisticated computer video games — the ability to work with all S-100 bus products.

Full expansion capability

Tailor the Sol-20 system to your applications with our complete line of peripheral products. These include the video monitor, audio cassette and digital tape systems, dual floppy disc system, expansion memories, and interfaces.

Write for our new 22 page catalog. Get all the details.

Processor Technology, Box C, 6200 Hollis St., Emeryville, CA 94608. (415) 652-8080.

Processor Technology Corporation

FCC Grants Tymnet Carrier License; Biggest 'Smart' Net Now Open To All

The Federal Communications Commission has granted approval for Tymnet, Inc. to operate as an interstate communications common carrier, clearing the way for any organization nationwide to access TYMNET, the largest and the most intelligent value added network chartered as a common carrier.

Unlike other value added carrier networks which rely heavily on multiplexing equipment at the majority of network access points, TYMNET employs intelligent communications processors in all but three of its nodes which provide local dial access in some 80 cities across the country. This means that important user features possible only with intelligent processors — such as error control, alternate routing, and the ability to connect virtually any type of terminal device — are available to virtually all TYMNET users throughout the network.

Services authorized for Tymnet, Inc. in its new carrier charter include "reliable, low-cost connections over its network for terminal-to/from-computer, computer-to-computer, and, through store and forward techniques, terminal-to-terminal message communications." Customers in any business or endeavor may apply this range of services to such functions as data base access, general time sharing, collection of data from various locations, automated order entry, and transmission of business messages.

Immediately offering service in the low-speed range (up to 1200 bits per second), Tymnet will phase in other data communications services and its message switching service in the near future.

Tymnet, Inc., 10261 Bubb Road, Cupertino, California, 95014. (408) 446-7000. Regional and district sales offices are located in Cupertino, Houston, Rockville, Md., Atlanta, and West Newton, Mass.

MUSIC TO EAT CROW BY

Once again we failed to credit Gordon Miller, editor of *The Point Set* as author of "The Four Color Problem," Jan/Feb issue, page 82. Sorry.

put...input/output...in

Comparing The HP-25 with the SR-56

Dear Editor:

A couple of notes to James Blodgett's review, in your November-December issue, of the HP-25 programmable calculator:

The Texas Instruments SR-52 and the HP-65 are worth comparing, as he suggests, in that they both provide external storage of programs on magnetic card. However, the reader interested in making comparisons will want to compare the HP-*25* with the SR-*56*. The latter was not mentioned in the review, but it, not the SR-52, is the comparable item. HP-25 actual retail prices as I write this are around $135-$145; the SR-56 is running just under $100. Late winter seems to be the big price-drop time in calculators, so readers will be well advised to check again. Ads in the *The New York Times* are a convenient guide, but of course can't replace phone calls to local stores.

There is also the HP-25C, which "remembers" the last program keyed in even when turned "off". Of course, it costs more.

Comparing the Hewlett-Packard and the Texas Instruments calculators at either of the two levels (with magnetic card and without) is not a task to be taken lightly. There is no *direct* match of programs steps and keystrokes, not to mention how much you can accomplish in *n* number of steps. Just an example, which is revealed simply by consulting the marketing brochures: To put a number in a memory register while adding it to a number already there, keeping the sum, the Hewlett-Packard routine of STO + *n* consists of three keystrokes, but uses *one* program step. By contrast, the SR-52 ad shows "STO 15" as using *three* program steps. Presumably, the addition would have to be done first, since the SR literature does not mention what HP calls "register arithmetic" — the capacity to operate arithmetically on the display and a register, and store the results in that register. To be fair, and further compound the contrasts, the Texas Instruments devices boast "indirect storage" (put a number in the register whose address is stored in the register addressed), a stunt the HP family discussed here does not do. A shopper should also notice that "GO TO STEP nn" takes one program step for Hewlett-Packard but requires three by Texas Instruments.

You cannot, then, determine the length of a program, which these machines will accept by looking at a sentence which states the number of program steps each device allows. No doubt a reviewer who embarks upon a careful comparison of the two calculators (and, of course, that was not Blodgett's purpose) will find numerous more subtle differences. I have cited only those which seemed important to me, for the kinds of programs I wanted to write. I went to the HP-25 and, like Blodgett, became fascinated as well as pleased with how much I could do in "49" steps. Other readers may find indirect addressing and flags, features of the Texas Instruments line, overwhelmingly attractive. There have been a few times when I wish I had them, but only once while writing some fifty programs for the HP-25 in the ten months I've owned one, have I been utterly blocked by a problem where I *thought* the TI program might resolve it. Of course, you can't be sure unless you've got them both on your desk.

Another thought for anyone who'd like to buy both and review them for this magazine: The whole business of "stack" vs. "algebraic" hierarchy is probably more important than the nit-picking concerns I've demonstrated above. It's just that I hesitate to pronounce upon the subject since I'm now thoroughly steeped in "stack-think." I went back to my four-banger Litronix the other day (a model with "parentheses") and it momentarily drove me crazy. But that's not fair, because I never quite had the hang of that system in the first place. And anyhow, it's not a TI. I *do* know that the ads do not resolve the issues; I don't envy the job of a copy-writer at either HP or TI when they undertake to convince us of the superiority of their own brand while impressing upon us the "objectivity" of the comparison. There could probably be a small book written on the topic. The ad writers on both sides do a remarkable job in the space they allot to this; the shopper will profit from reading these blurbs carefully. I suspect that the only interesting way to explore the question of how many program steps each needs to work is by writing a very large set of paired programs which embrace many different kinds of problems, then getting out some averages and standard deviations on the number of steps involved. The study should produce statistics in distinct categories of "kinds" of problems undertaken, or it will have little value. You can see why I haven't done it. It's just too big a job for the COSMEP publishing game, and where else are you going to "sell" a piece of writing like that?

Finally, an even "nittier" comment, but on something I overlooked for a while and evidently Blodgett did too: He is correct in saying that ten-to-the-power function is slightly off at seventh-and-higher powers. *But* if you are scaling, the user's manual for the HP-25 suggests ENTER, EEX, *n*, times, for that job. This, as far as I can tell, is *not* off. I think that ten-to-the-power ought to be confined to alogs, as a matter of habit, so that you won't inadvertently call in a slight error if you are working, for example, in "f FIX 2," where the oddity does not "show" below seven. (If you light up at f FIX 9, you'll see it at 5 and higher.)

George Thompson
2115 Luray Avenue - Apt. 2
Cincinnati OH 45206

Why pay $49 for a complete book on a new higher level language when it's worth more?

SCELBAL. A Higher Level Language for 8008/8080 Systems.
An unequalled book of solid documentation with room to grow.

You've heard about it. Now, it's available to everyone. **SCELBAL. SCientific ELementary BAsic Language.** A new high level microcomputer language for 8008/8080 systems that's simpler than machine language. This complete 368 page, profusely illustrated book is so comprehensive you might not be able to use all its information for years. But as you and your system mature, this book will stay forever young with refreshing ideas and extensions of your own parameters.

A Total Language, including Source Listings

Here's everything you need. A gold mine of routines, techniques, source listings, flow charts and lots more. It took 2½ years to compile, evolve and write. Now it's yours for years to come. Look at all you get!

5 Commands: SCR, LIST, RUN, SAVE, LOAD. **14 Statements:** REM, LET, IF . . . THE, GOTO, FOR with STEP, END, INPUT, PRINT, NEXT, GOSUB, RETURN and optional DIM. **7 Functions:** INT, SGN, ABS, SQR, RND, CHR, TAB. And, it runs in 8K and more.

Contains Overall Logic Description Program

The first eleven chapters, over 160 pages, describe the statement interpretation logic of the program, using flow charts, commented symbolic assembly language code and verbal descriptions. The book also details SCELBAL 8008/8080 assembly, providing absolute code for the program in octal, beginning at origin 01/000 for either version. Plus, you'll find painstaking operating instructions and even suggestions for tinkerers and innovators.

Software is Our ONLY Business

We know there's no one program for **every** potential user. So, particular care has been taken to provide a good, sound, fundamental selection of syntax statements and functions in this language. You get all the information necessary to get started . . . and to grow and expand with ease as you and your system capability evolve. The language is easily custom-implemented in virtually any 8008/8080/Z-80 based computer!

Free Bonus "SCELBAL UPDATE" Included!

Every copy of SCELBAL may be registered with the publisher. Registration assures that you'll receive the first 6 issues of "SCELBAL UPDATE" absolutely free. These exciting supportive improvements and suggestions are combined with user's unique findings and ideas. "SCELBAL UPDATE" is a LIVING concept. As you and your system grow, you can update at will. We've only scratched the surface. There's more excitement to come!

Introducing 2 New SCELBAL Supplements!

SCELBAL has taken off so fast, two special supplements had to be printed. First, there's **Extended Math Functions:** SIN, COS, LOG (BASE E), EXP (BASE E) and ATN . . . only $5.00 ppd. The second supplement, **String Handling Capabilities,** includes the numeric functions LEN and ASC. It sells for only $10 ppd.

Don't delay. Order SCELBAL today!

You can own this invaluable book for only $49 ppd. You'll have all the data you need to custom-tailor a high level language for your system. The revealed secrets of its operation are virtually priceless. "SCELBAL — A Higher Level Language for 8008/8080 Systems". Only $49 ppd. And, it includes six (6) Free "SCELBAL UPDATES" too. Order today!

Here are typical comments about SCELBAL . . .

"SCELBAL looks fantastic! My system isn't even on-line yet, but I'm eagerly looking forward to activation day."

"Your section on statement interpretation taught me to understand how a high level language operates. I never thought I could do that before!"

"I've tailored SCELBAL software to my own insurance business. Now I offer new, expanded computerized service to my customers, using my own 8008."

SCELBI COMPUTER CONSULTING INC.
1322 Boston Post Road Rear
Milford, CT 06460/(203) 874-1573

Prices shown for North American customers. Master Charge, Postal and Bank Money Orders preferred. Personal checks delay shipping up to 4 weeks. Pricing, specifications and availability subject to change without notice.

put... input/output... in

Changing/Dates

Dear Editor:

It has been brought to my attention that certain days for particular dates in my article "Days and Dates" (November-December, 1976) are in error. Mr. Jerome Greene, Market-Math Incorporated, is correct in his computations and corrections that the days for questions 2, 4, and 13 should be corrected.

Q2: change the day for July 4, 1776 from Sunday to Thursday;

Q4: change the day for April 14, 1865 from Tuesday to Friday; and

Q13: change the day for June 25, 2950 from Monday to Sunday.

It is indeed encouraging that there are interested readers of *Creative Computing* who become so involved in the articles that they find the mistakes of the writers. It is more encouraging that the readers will, with all respect, communicate their results to the writers.

My sincere thanks to Mr. Jerome Greene for his corrections.

James Reagan, Jr.
23234 Sherman
Oak Park, Michigan 48237

Dear Editor:

In the article, "Dates and Days," (November/December 1976) there were several errors. The equation for the Zeller's Congruence should be given as:

F=INT(2.6*M-0.2)+k+d+INT(D/4)+INT(C/4)-2*C)mod 7

Another error was in the example. The absolute value marks indicating integer function "INT" were omitted in the first equation. They were later included.

Another error was in the answers to the illustrative quiz. For the second date given, July 4, 1776, the correct F value is 4; for the fourth date given, April 14, 1865, the correct F value is 5; and for the thirteenth date given, June 25, 1950, the correct F value is 0. These values were checked against Encyclopedia Britannica Micropedia Volume II, page 455 in the article, "Perpetual Calendar."

It was not mentioned that the Zeller's Congruence given is only valid on and after October 15, 1582 when the present Gregorian Calendar begins. Prior to that date, there was the Julian Calendar, which is valid on and before October 4, 1582. Between those two dates, several days were skipped to correct the calendar. I would be interested in knowing if anyone has the Zeller's Congruence for the Julian Calendar.

Michael A. Smith
273 South Third Avenue
Ilion, NY 13357

Fortran Users Arise!

Dear Editor:

Since I run on an IBM 370 with TSO, and FORTRAN is the only useable time-sharing language on our system, I cannot use either BASIC, APL, or punched paper tape. I am therefore very interested in any material that might be available in FORTRAN on punched 80-column cards or on ½ inch magnetic tape. If information on what programs might be available in this form exists I would greatly appreciate seeing it in some future issue of *Creative Computing*.

I have running now working FORTRAN IV versions of the Startrek and ROCK[1], translated more or less verbatim from the BASIC versions in "101 Basic Computer Games." Startrek has a couple of minor improvements, such as the ability to escape into hyperspace.

Roger D. Deschner
University of Chicago
Computation Center
5737 University Ave.
Chicago, IL 60637

Ed. note: Readers — can you help?

Cover Copy By Computer

Dear Editor:

The cover of *Creative Computing* for September/October 76 intrigued me, to put it mildly. I was determined to do it on a computer - here is the result. It turned out to be quite trivial.

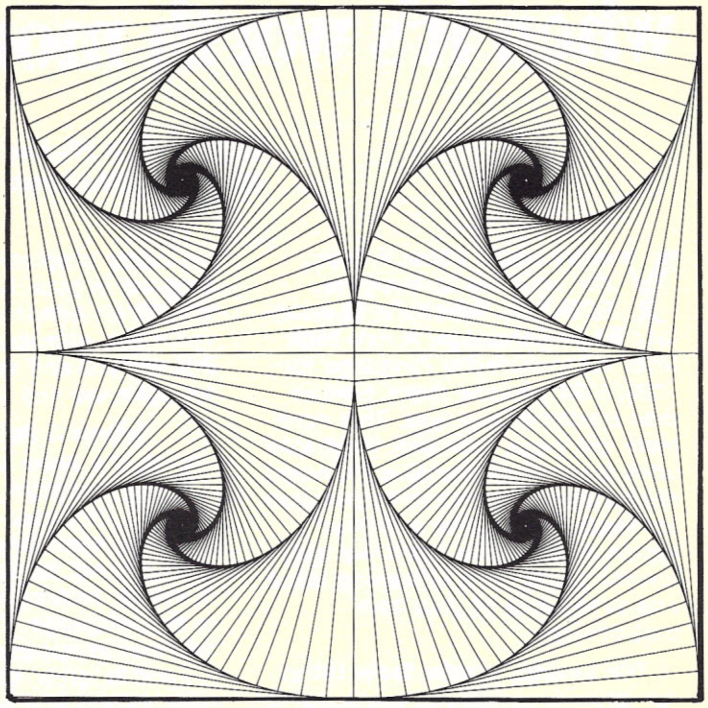
The program, written in FORTRAN-IV PLUS, ran on a PDP 11/45 which created a plot tape, which was read by a PDP 11/10 running a CalComp 960 plotter.

I have a problem — the FORTRAN library exponentiation routine here will not take a floating negative number to a power. Do you or any of your readers have a suitable algorithm which will avoid this problem (there is no problem with fixed values, only floating)?

One more comment: *Creative Computing* is great, keep up the good work.

Samual Henning
Box 1140
Timmins Ont.
Canada P4N 7H9

Editor's note. Send your algorithm to Sam and to us for publication.

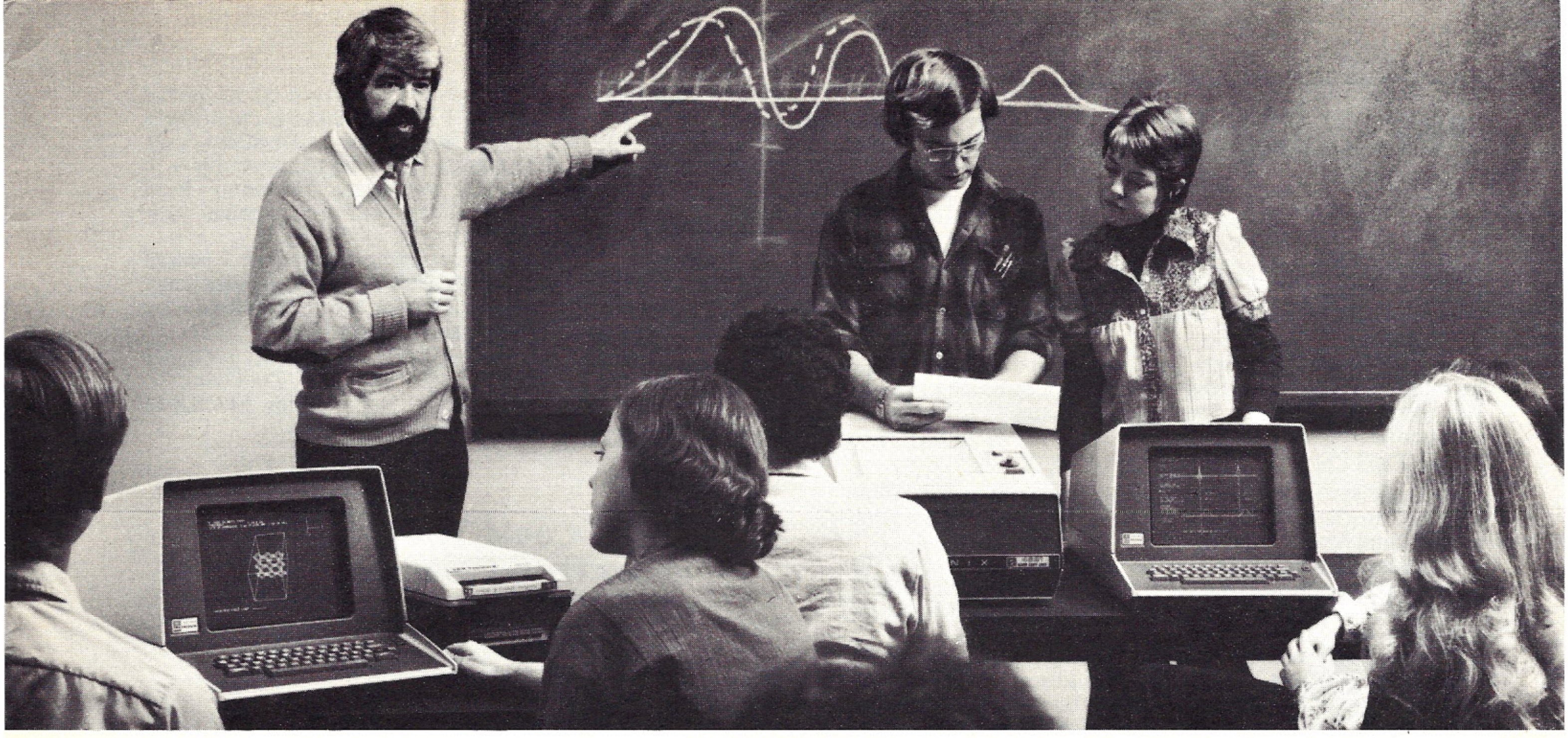


Great Feets

To the Editor:

In response to reader Rowlands inquiry in the Nov-Dec 1976 issue of *Creative Computing,* I would like to mention some research now being undertaken by the Personal Computing Group of Flakey Systems. In the Northern hemisphere we have found that the "head-to-head" transmission described by Rowlands occurs in the feet. The "gassing" described by Rowlands is reproduced by applying sweaty socks inside old sneakers and jogging. The communication is quite subtle but effective to human olfactory systems. The frequency depends on physical conditioning of the human, not upon any external signs. Likewise the best broadcasts occur when one is in danger or in competition. Our medical group finds no side effects from this communication. I will look forward to when our American feet will collectively communicate with the Australians' heads.

Mario DeNobili
Central Park, NY



Graphics good enough for research...

now is priced right for the classroom.

The 4006-1 from Tektronix: laboratory, time machine, testing ground . . . all for the price of alphanumerics.

Words alone don't always suffice. Very often the subject demands a graphic demonstration. Tektronix' affordable new 4006-1 lets you evolve to computer-aided Graphics demonstration right in the classroom. It's the same high-resolution Graphics on-campus research has relied on for years.

Unlike mere graphing terminals, the 4006-1 offers enough information density to take your students into dynamic, 3-dimensional detail. Enough to let them experiment with the properties of lethal radiation. To observe genetic mutations over generations. To test architectural structures against stress and strain. They can change parameters at will, or translate massive printouts into conceptual graphs.

It's a package that assures maximum flexibility: our proven interfacing and software connect with virtually all mainframes. Our new Interactive Graphing Software Package lets anyone graph with conversational ease. You get plug-to-plug peripheral options like the 4631 Hard Copy Unit that serves up to four terminals; or the 4923 Digital Cartridge Tape Recorder that takes program storage off-line. And you get Tektronix' reputation for research equipment, educational expertise, and worldwide maintenance support.

Single unit price: $2995, with generous quantity discounts. Ask your local Tektronix Sales Engineer how your discipline fits into the picture. Or write,

Tektronix, Inc.
Information Display Group
P.O. Box 500
Beaverton, Oregon 97077

TEKTRONIX®

†U.S. Domestic price only
OEM prices available

put...input/output...in

Letter From the Future With A Message For The Present

Dear Editor:

(The quotation from the Encyclopedia Galactica here reproduced is taken from the 117th Edition published in 1030. F.E. by the Encyclopedia Galactica Publishing Co., Terminus, with permission of the publishers.)

A SACRED COW CALLED EFFICIENCY

As the Second Foundation began to prosper there were some problems in the Galaxy. Inter-planetary transportation took too long and was unreliable.

A special Council of the Speakers was held on Trantor. The Council decided to build a new star-travel system and hired Interstellar People Movers, the largest corporation in Space, to do the job.

"We want a system that will allow us to go quickly and easily from planet to planet," the council told Interstellar.

Interstellar dispatched armies of workers who went to work immediately. Only the most modern equipment was used — Molecular Nebulizing Phasers, Deuterium-Oxygen Space-warpers, Transcendental Bulk Mobilizers and more.

At enormous cost, Interstellar produced what was hailed as "the most efficient transportation system in Space."

The day after the new system was opened the Council of Speakers called in the Interstellar designers for a hearing.

"Our system can move a person one thousand parsecs in one minute and has a basic cycle time of only one hundredith nanosecond," said the designer.

"That's true," said the Council, "BUT IT ONLY GOES TO ONE PLANET!" (It seems the system connected Trantor to a single planet on the Periphery, Terminus. Almost no one wanted to go to Terminus and even those who did had difficulties: to ride the new system they had to use paper cards called "Jargon-card Labels." These labels had to be punched with cryptic messages like "/&" or "**eoj" and since users couldn't understand what the messages meant, they usually punched their cards the wrong way and spent a lot of time trying to find the mistakes.)

"Suppose," asked the Council, "that we want to go to Chemyllas or Terra or any of the other planets in the Galaxy?"

"You can't DO that on our system," said the Interstellar People Movers people, "but if you want to go to Terminus, our system is the most efficient transportation in Space."

Moral: Efficiency without proficiency is not productive.

J.P. Peer
417 West Water St.
Berne, IN 46711

Creativity Is and Isn't

Dear Editor:

Your most recent issue (Nov-Dec) of *Creative Computing* urged me to sound off a bit. I really wanted to, last issue, but didn't, eventually. I have been aware of *Creative Computing* since its second issue, and I want you to know I believe in and support what you are doing. With every issue I get the distinct impression that "That Ahl character is trying to tell us something." Last month's feature on computer poetry drew me out, and this month's "Computers and Beauty" boiled me over. Something has to be said. May I? Creativity is not a computational process.

The creative act is the act of producing something that wasn't there before. To produce something from nothing. And creativity is something we use everyday; writing a sentence on paper, a new sorting algorithm, discovering a new, shorter way to walk home from school. Creativity is arranging words and notes together in a new way that expresses a feeling that heretofore went unsaid. Creativity produced the computer, the microprocessor, and Star Trek. I feel the Messers. Sasaki sadly shortchange their own self-realization when they say "we must implement, more or less, the creative power to computers." Please, please stop thinking creativity is merely a mathematical function, a process, a technique! The technique of creativity is no technique at all. Or, to put it another way, all techniques are aspects of creativity. You can no more define creativity by a listing of techniques than you can define "television" by a listing of its components. Creativity is *what's left* when your technique is defined.

I became so emotional by the article that maybe I missed their point altogether and we are actually both saying the same thing. I will write them too in a day or seven. Thank you for your ear. Here's a poem, and a survey form too.

Lionel J. Goulet
236 Union St.
Millis, MA 02054

P.S. I love the Winograd article!

What Could You Do With Your Own Computer?

If you had your own computer, you could ...
Balance Your Checkbook
File your Cookbooks
Simulate Spaceflight
Win a Dogfight
Watch your stocks
Eliminate your clocks
Define words
Classify birds
Give first-aid
Get better grades
Christmas Cards
Aren't even hard
Daily reminders
Antidote finders
Inventory Parts
Graphic Arts
Newspaper type
Conway's *Life*
Command a starship
Improve your golf grip
Mastermind
Nursery Rhymes
Bookkeeping
You've been sleeping
If you can't find
Inside your mind
Something to do
With your own Compu
Ter.
The question "What could ...?"
Has too many answers
For poetic listings
And numerical dancers
Besides, it too easy.
Anything could.
Let's tackle the tougher
Tell me, "What *would* ...?"

Altair Editor Anybody?

Dear Editor:

I am interested in using my Altair 8800 to produce clean letters and reports from text entered through a keyboard with editing capabilities. Is there anyone out there who has an Altair compatible software package that can help me?

Ludwig Braun
Professor of Engineering
State University of New York
Stony Brook, NY 11794

Ed note: I'd like to find a package like this too for either the Altair 8800 or SWTPC 6800. The only one I know of currently is an 8080 package which appeared in *Dr. Dobbs Journal* in the June/July 1976 issue. /DHA

al... editorial... editor

Fred Hofstetter, President of the National Consortium for Computer Based Musical Instruction, and a music professor at the University of Delaware, is the special guest editor for this issue of *Creative Computing.* His excitement about the possibilities of using the computer with music convinced us to let him share some of the accomplishments in this area with you, and with seemingly unlimited energy, he recruited not only the diverse, informative articles that appear on these pages, but scores of others, for which we just couldn't find the room to print. I asked him to describe the scope of these articles; this is found in his guest editorial.

One unusual observation: over and over I've encountered a great interest in music among the same people who are interested in using computers innovatively. I don't know the reason for the connection, just that it exists. The wealth of information pooled here as the results of Fred's efforts should answer the questions many of you music lovers have about what you can expect from the union of computers to music.

Dan Burgoon of Solid State Music recruited one article on hobbyist computer music for us; we'd like to run more articles on homebrew computer music systems in future issues as space permits. Send us information about what you're doing.

I've done some statistical analysis of the first of the surveys you sent in from the November/December issue (a full report on the Surveys will appear next time), and one fact is evident: you look to *Creative Computing* for diversity! Hence, we've tried to provide a variety of topics in addition to the music features.

Dave Ahl and I plan to do quite a bit of traveling these next few months, to attend a number of conferences around the country. You can have even more impact in determining the direction of *Creative* by stopping by the *Creative Computing* Booth at the conference in your area, introducing yourself, saying "I'm a subscriber and I think. . . ." Look for one or both of us at The First West Coast Computer Faire (Booth 214) April 15-17 in San Francisco; ACM, Biloxi, Mississippi, April 18-20; National Computer Conference, June 13-16 in Dallas (Booth 71), The Personal Computing Fairs in Philadelphia on April 30-May 1, and Boston, June 18-19, the Trenton Fair in Trenton, N.J. of course, April 30-May 1, and The Atlanta Computerfest, June 18-19 in Atlanta, Georgia.

—Burchenal Green

Guest Editorial

Since the late 1950's, musicians have been using the computer to solve a variety of problems related to the composition, analysis, synthesis, printing, performance, and learning of music. Never before has a single computational device had such universal appeal to musicians. For this issue of *Creative Computing* papers have been invited from the most active developers of music programs. The first two articles concern the composition and sound-synthesis of music. Gary Nelson of the Oberlin Conservatory of Music describes MPL, a *M*usic *P*rogram *L*ibrary which contains routines for the generation and analysis of musical patterns. Sidney Alonso, Jon Appleton, and Cameron Jones of Dartmouth College describe a stand-alone, dedicated minicomputer which combines generative routines and a sound synthesizer in the same system.

Then Thomas Whitney of The Ohio State University explains how encoded music can be analyzed by means of SLAM, a *S*imple *L*anguage for *A*nalyzing *M*usic. The goal of this and similar efforts is to encode a large data base of musical compositions, and through computer-based analytical techniques, to generate a comprehensive body of knowledge about the creation and evolution of compositional styles in music.

The most recent application of computers in music has been the advent of computer-based musical instruction. Hardware advances have made it possible for computers to perform, display, and ask questions about music. Efforts to realize the potential of computer-based musical instruction are being made by over one hundred developers. In my article I explain why musicians are so excited about computer-based musical instruction.

During the 1960's many humanists opposed the use of the computer for musical purposes. They feared that computer techniques would lead to a depersonalization of the arts. Looking back at what has happened in the last decade, one can see that the computer has actually helped musicians get closer to their art. In his article, humanist Robert Taylor of Columbia University shows that there are many similarities between programming computers and directing musical productions, and he suggests that much can be learned by studying these similarities. And if you want to get personally involved, there are a few music programs written in BASIC which might be fun to try. Just want to listen? John Selleck of the Massachusetts Institute of Technology has provided an annotated guide to commercially available recordings of computer music. Want more reading? Then turn to the reviews of books dealing with computer applications to music.

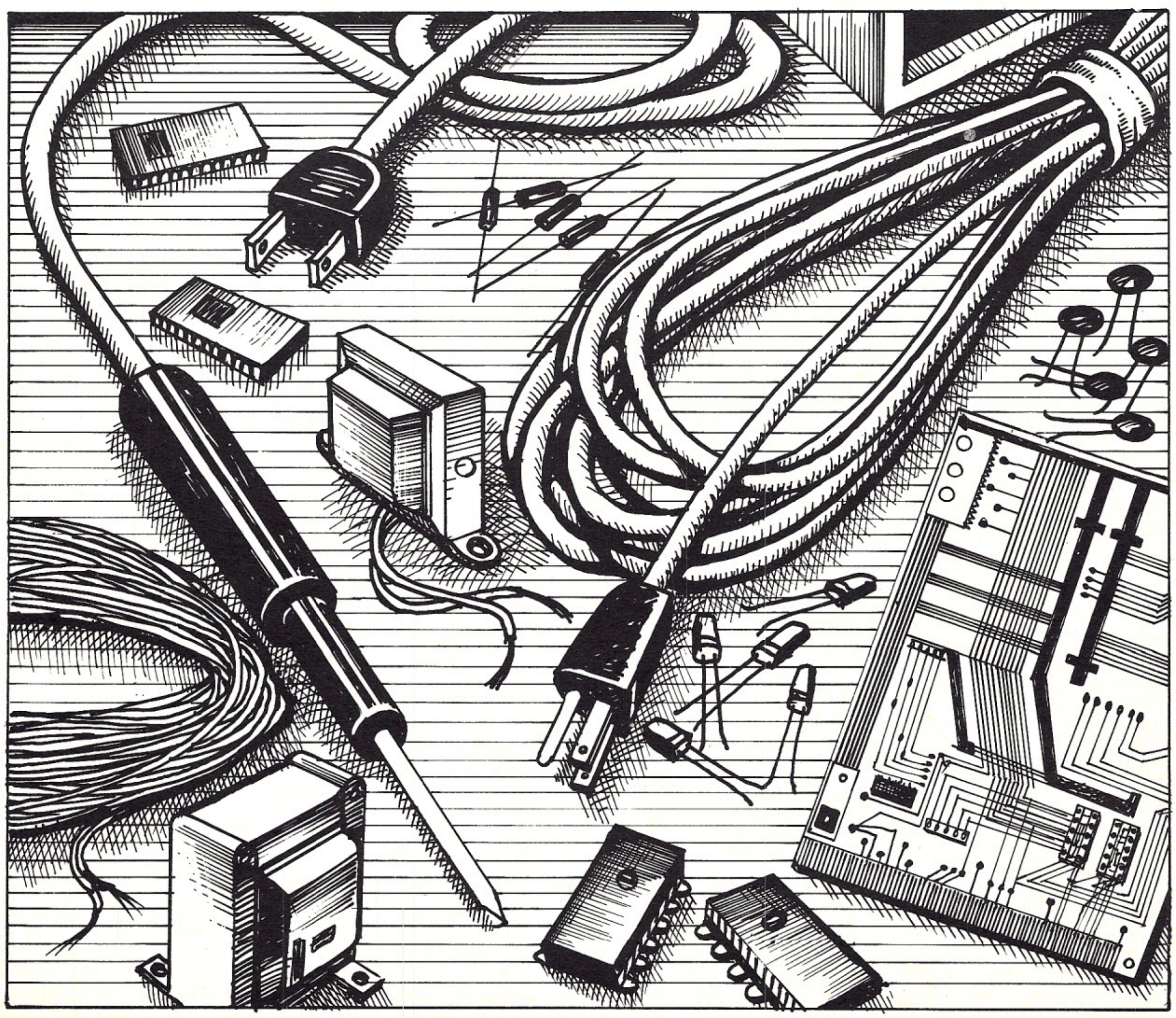
—Fred Hofstetter

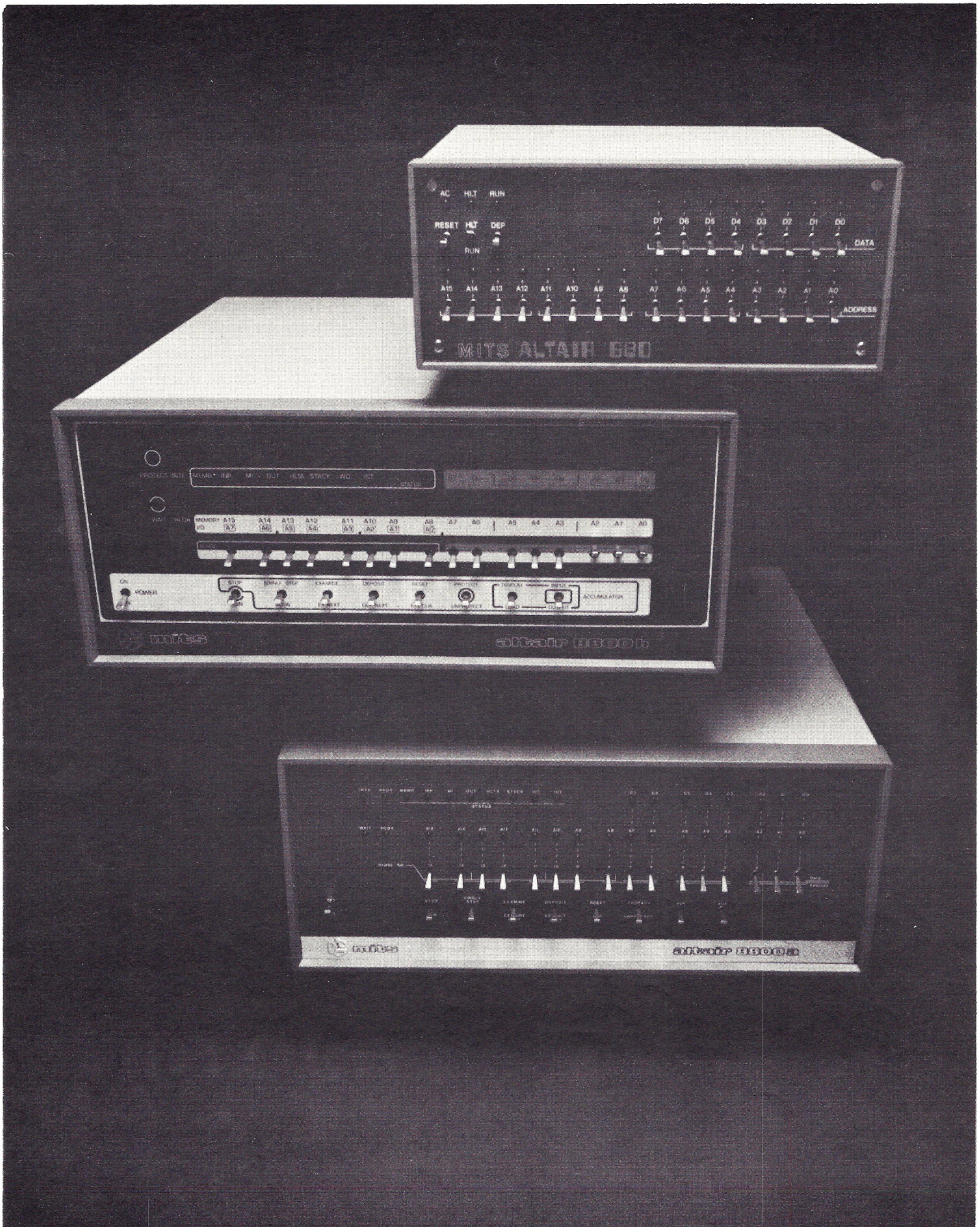
Musical Score Printed by a Computer at Stanford University's Artificial Intelligence Laboratory using Professor Leland Smith's music printing language.

KIT-A-MONTH

The Altair™ kit-a-month plan allows you to own an Altair mainframe without taxing your pocketbook. Mits has made it easy for you to purchase an 8800a, 8800b or 680b computer in monthly installments where you receive components with each payment. There are no financial charges because we have made each monthly shipment a kit in itself. This will give you time to read up on computers and/or gain knowledge from friends.

We have set up an Altair kit-a-month payment desk to service your needs. When writing or calling the factory for information about your shipment or account, just refer to the "kit-a-month payment desk."




AC HLT RUN
RESET HLT DEP
RUN
D7 D6 D5 D4 D3 D2 D1 D0
DATA
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
ADDRESS
MITS ALTAIR 680
mits
altair 8800b
mits
altair 8800a

OWN YOUR 680b IN FIVE EASY PAYMENTS

1st month	680b Manuals, Main and Display PC Boards, and membership in users group
2nd month	680b Case and Power Supply
3rd month	Parts for 680b Display Board
4th month	Parts for 680b Main Board less Integrated Circuits
5th month	680b Chips

$93.20 plus $2.00 per month make your payments $95.20 per month and you have your complete kit in five easy payments. This plan gives you the full 680b with memory and I/O. The BAUDOT Option is still $42.00 extra and may be purchased with Time Payment #4 or #5. Remittance of $95.20 will start you on your way to owning your own computer.

Alaska, Hawaii, APO and FPO customers include $3.00 per month for Air Parcel Post charges instead of $2.00 for regular Parcel Post shipment. This would make your payment $96.20 per month.

Canadian customers include $3.00 per each month for postage and handling fees.

NOTE: This plan does not apply to foreign sales other than Canada.

NEW ALTAIR 8800B

Available on Time Payment Plan

$105.00 payment per month plus $2.00 postage and handling for each kit makes an easy $107.00 per month to own the newest of the Altair processors.

Send in the first $107.00 payment and you will start receiving your 8800b Kit as soon as we receive your order.

8800b Time Payment #1	8800b Manuals and Users Group Membership
#2	EC-18
#3	Power Supply Board & Parts
#4	Transformer
#5	Display Board & Parts
#6	Case
#7	Main Chip, Buffer Board & Parts
#8	CPU Board & Parts Less 8080A

Alaska, Hawaii, APO and FPO customers please include $4.00 for shipping charges (making $109 per month payments) for Air Parcel Post shipment. Otherwise, shipment will come Parcel Post, not insured.

Canadian customers must accept month #6 Emery Airfreight Collect. All other months must include $4.00 postage and handling making monthly payments of $109.00.

$79.00 / Month
ALTAIR 8800A TIME PAYMENT PLAN

8800A Time Payment #1	8800A Manuals and Users Group Membership
#2	EC-18, PC Board and Hardware
#3	8800A Power Supply Kit
#4	8800A Case
#5	CPU PC Board and Bag of Parts less the main chip
#6	Main Processor Chip
#7	Display Control Board and Parts

The price of the Altair 8800A mainframe is $539.00. Seven easy payments plus $2.00 per month for postage and handling charges make this plan equal $79.00 per month. Upon receipt of your first $79.00 payment you are on your way to owning your own 8080A basic computer system. A list of available compatible peripherals is enclosed to let you plan your system as you learn about your microprocessor. By 8800A Time Payment #7 you're ready to go.

Alaska, Hawaii, APO and FPO customers please include $4.00 for shipping charges (making $81 per month payments) for Air Parcel Post shipment. Otherwise, shipment will come Parcel Post, not insured.

Canadian customers must accept month #4 Emery Airfreight Collect. All other months must include $4.00 postage and handling making monthly payments of $81.00.

KIT-A-MONTH

ORDERING INSTRUCTIONS

Feel free to contact your local Altair dealer (as listed on CREATIVE COMPUTING's back cover) for ordering details. If there is no Altair dealership in your area yet, follow the steps below to expedite Kit-A-Month orders that are placed directly with the factory.

1. Send all payments other than BankAmericard or Master Charge in the form of a cashier's check or money order. Personal checks are acceptable, but clearance time will delay your order by 2-3 weeks.
2. The kit-a-month plan has been set up to proceed in order and we cannot deviate from that order. You can help us by noting with your payment what month you are on.
3. When calling or sending in orders, refer to your customer name on the original order and also your Mits order number.
4. If you change your address, keep your name as it is on the original order to keep records straight.
5. Please note special instructions for Alaska, Hawaii, APO, FPO and Canadian customers. If these are not followed, it could result in delays in processing your order.
6. The Kit-a-Month desk has been set up to help expedite your orders because of the overwhelming response we've had with previous time payment plans. Please feel free to use this service whenever you have questions. When writing letters to Mits, simply note "Kit-a-Month desk" on the outside of the envelope.

NOTE: Once you start the Kit-a-Month plan you are guaranteed the existing price at the time of your first order. You will not be affected by price increases.

Enclosed is my payment of ____________ for the first shipment of my Altair kit-a-month.

Master Charge #____________________ or BankAmericard #____________________

Altair 680b☐ Altair 8800a☐ ☐Altair 8800b

NAME____

ADDRESS____

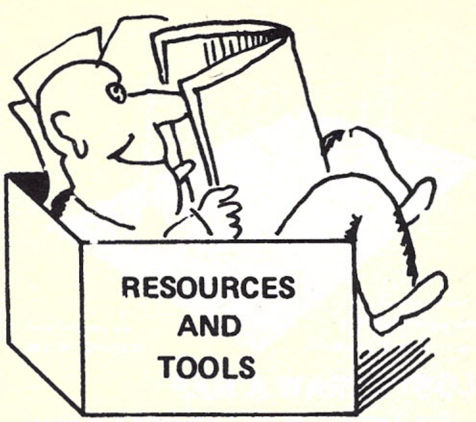
CITY_____________________________ STATE & ZIP______________

mits/2450 Alamo SE/Albuquerque, NM 87106 505-243-7821

Prices, specifications, and delivery subject to change.

mits/2450 Alamo SE/Albuquerque, NM 87106 505-243-7821

COMPLEAT COMPUTER CATALOGUE



We welcome entries from readers for the "Compleat Computer Catalogue" on any item related, even distantly, to computers. Please include the name of the item, a brief evaluative description, price, and complete source data. If it is an item you obtained over one year ago, please check with the source to make sure it is still available at the quoted price.

Send contributions to "The Compleat Computer Catalogue," *Creative Computing*, P.O. Box 789-M, Morristown, NJ 07960.

BOOKS AND BOOKLETS

COMPUTER MUSIC JOURNAL

Devoted to the development of computer systems capable of producing high fidelity music, this new journal will cover such topics as: computer composition of music; design of real time playing instruments; real time input controllers such as Keyboards, joysticks, and new controllers, homebrew computer music instruments.

The first issue of the journal will be about 50 pages in length, and will increase in size and scope as interest dictates. The first issue should be out by now. Bimonthly $14.

Computer Music Journal, PCC, Box 310, Menlo Park, CA 94025.

INTO EVERY LIFE A LITTLE RAIN MUST FALL

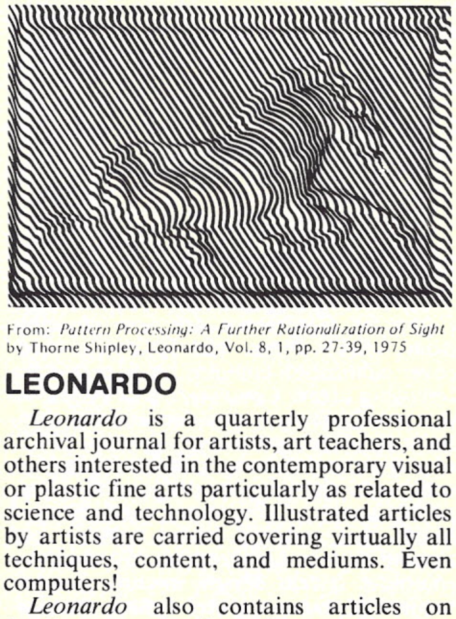
And if the rain is *Rain* magazine, consider yourself fortunate indeed. *Rain* is a lively publication which has for a subtitle "The Journal of Appropriate Technology." It runs about 24 pages per issue and carries mainly resource notes (similar to the Catalogue section of *Creative*) on a wide variety of subjects such as agriculture, architecture, education, energy, tools, health, soft technology, communications, food and nutrition, yes, and even computers. *Rain* started off with a bias toward the Pacific Northwest but today covers all of North America quite well. Annual subscription of 12 issues $10. "Living lightly" subscription rate is $5 ("living lightly" means you *really* can't afford $10). Sample issue $1.

RAIN, 2270 N.W. Irving, Portland, OR 97210. (503) 227-5110.

CRYPTOLOGIA JOURNAL

CRYPTOLOGIA, a journal devoted to all aspects of cryptology, will be published four times a year and will contain research papers, survey articles, personal accounts, reviews, educational notes, and problems. Some of the areas which will be discussed are mathematical, computational, literary, historical, political, military, mechanical and archeological aspects of cryptology. Editors are: Cipher A. Deavours, Department of Mathematics, Kean College; David Kahn, Department of Journalism, New York University; and Brian J. Winkel, Department of Mathematics, Albion College. Quarterly, $16/year.

CRYPTOLOGIA, Albion College, Albion, Michigan 49224.



From: *Pattern Processing: A Further Rationalization of Sight* by Thorne Shipley, Leonardo, Vol. 8, 1, pp. 27-39, 1975

LEONARDO

Leonardo is a quarterly professional archival journal for artists, art teachers, and others interested in the contemporary visual or plastic fine arts particularly as related to science and technology. Illustrated articles by artists are carried covering virtually all techniques, content, and mediums. Even computers!

Leonardo also contains articles on developments in the other arts, on new materials and techniques, and on subects in aesthetics, architecture, education, the natural and social sciences, and technology. Extensive reviews section.

Subscriptions for institutions $55 per year, for individuals $15.

Pergamon Press, Maxwell House, Fairview Park, Elmsford, NY 10523.

The publisher writes that illustrated manuscripts are being sought on all aspects of computers in art. For information on submittal requirements, write Dr. Frank Malina, Editor, *Leonardo*, 17 rue Emile Dunois, 92100 Boulogne sur Seine, France.

UNEARTHLY WONDERS

UNEARTH is a new science fiction magazine devoted to publishing stories from new writers. It runs the entire gamut — speculations, fantasy, horror, computers, robots, etc. *UNEARTH* also includes book and movie reviews, a science column by Hal Clement, and a special feature from contributing editor Harlan Ellison (in issue #1 this was Harlan's very first story, "Glowworm"). *UNEARTH* is also seeking stories and artwork. Published quarterly, $3.50 per year. Sample copy $1.00.

UNEARTH, Suite 190, 102 Charles St., Boston, MA 02114.

BOOKS AND BOOKLETS

CATALOG OF THE FUTURE

A huge 40-page tabloid catalog of books, magazines, learning materials, games, and cassettes is now available from the World Future Society. Catalog lists books by major and not-so-major publishers. There's something for everyone here — topics include computers, business/industry, communications, education, environment, forecasting, simulation/gaming, systems design, and many more. A worthwhile catalog and the price is right — free.

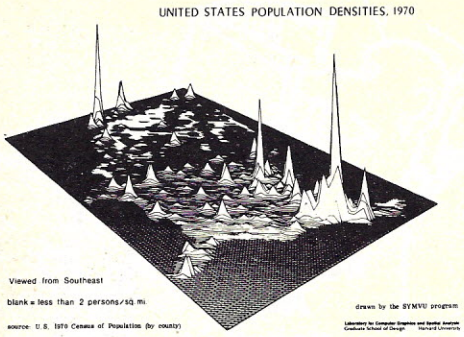
World Future Society, 4916 St. Elmo Ave., Washington, DC 20014.

COMPUTER-RELATED CRIME

Six new reports on computer abuse, system vulnerabilities, applicability of Federal and state penal laws, proprietary rights in software, and privacy law effects are available from the SRI Project on Computer Abuse headed by Donn B. Parker. Announcement of reports available and order form are available from Allison Brandt, Report Production, Bldg 300 B5, SRI.

Donn Parker is also interested in information about specific computer-related crimes and other abuse (vandalism, property theft or fraud, financial theft or fraud, unauthorized use of services, etc.). Write him at SRI or call (415) 326-6200 X2378.

Stanford Research Institute, 333 Ravenswood Ave., Menlo Park, CA 94025.



LOST? DRAW A MAP

The Laboratory for Computer Graphics and Spatial Analysis within the Graduate School of Design at Harvard University has just released a new edition of LAB-LOG, its catalog of computer programs, data bases and publications. Research at the Laboratory is principally concerned with the analysis and graphic display of geographic data used in the planning process. LAB-LOG describes various products which have resulted from this work and which are currently available for distribution to universities, government agencies and private organizations.

LAB-LOG includes a description of six different computer programs for use in the graphical display of spatial data via a line printer, line plotter and cathode ray tubes. The programs are written for IBM 370 series computers and range in cost from $575 to $1200. A wide variety of cartographic (x-y coordinate) data bases are also described. Publications are available on the subjects of automated cartography, theoretical cartography and theoretical geography. Publications cost from $1.00 to $12.00. LAB-LOG also contains a brief description of the Laboratory's history, research directions and operating policies. Copies of LAB-LOG $1.00 prepaid.

The Laboratory for Computer Graphics and Spatial Analysis, 520 Gund Hall - Harvard University, 48 Quincy Street, Cambridge, MA 02138.

MACHINE TALK

A brief look into the world of data communications is presented in this booklet. Explains in a simple way machine to machine and machine to people communication. Illustrates various types of data communications such as transmission of electrocardiograms and telephone to computer links. Free from your local Bell Telephone Co. Business Office.

TRANSISTOR AGE

This little booklet gives a very comprehensive explanation of the transistor and its tremendous impact on society. Describes the research leading to its invention, its operation, its applications and contributions to science and engineering. Free from your local Bell Telephone Co. Business Office.

ATSU

The Association of Time-Sharing Users is a professional organization whose purposes are to supply current information to time-sharing users, to provide information about the various products and services offered by remote computing suppliers. to provide a forum for discussion of topics pertaining to remote computing and interactive time-sharing. It provides four publications: a newsletter, a Press Review, "Interactive Computing," and "Interactive Computing Directories." Membership $20.

Hillel Segal, President, ATSU, 75 Manhattan Drive, Boulder, Colorado 80302

DIGITAL COMPUTER PLOTTING

This 64-page book by Franklyn K. Brown is a very nice introduction to the programming and use of a digital incremental plotter. All of the routines and programs in the book are for batch Fortran IV (CDC Cyber 72) and rotating drum plotter (Calcomp 565). The principles presented are explained in ample detail with lots of examples and diagrams. The author covers basic point-to-point plotting, symbols, scaling, axes, curved surface plots, dimensioning, and sample applications. Naturally if you have a flat bed plotter or a language other than Fortran, you'll have to make mental conversions to use this book, nevertheless the basic principles are still sound.

Franklyn K. Brown, Room 203 GR, College of Engineering, Northeastern University, Boston, MA 02115.

NCC PROCEEDINGS

The Proceedings of the 1976 National Computer Conference are now available from AFIPS Press. The 1082-page hardcover publication contains 136 papers in the following areas: *Computers and People* — computer privacy, computer security, computer abuse, computer cryptography, EFTS, education and training, computer graphics, computers and the physically handicapped, public access to computers, medicine and health care, criminal justice systems, and computers in architecture; *Systems* — computer system design, microprocessors, minicomputers, computer system management and planning, computer system performance and evaluation, computer networking in the U.S. and Europe, word processing and office automation, computer-assisted manufacturing, and computer-controlled publication; *Science and Technology* — computer architecture, multiprocessor systems, data base systems, large-scale networks, programming languages, mathematical programming, software design and engineering, and artificial intelligence. $50 each (50% discount to members of AFIPS Constituent Societies).

AFIPS Press, 210 Summit Ave., Montvale, NJ 07645. (201) 391-9810.

SURE THINGS: DEATH, TAXES, AND LAWYERS

Are you associated with a data processing facility that keeps any kind of records on individuals? Are you developing or using proprietary software? Do you collect any kind of Federal, state, or local taxes? If you can answer "yes" to any of the above questions, then you probably should take a look at *Computer Law and Tax Report* edited by Robert Bigelow. This is a monthly 8-page newsletter that covers legal and tax matters related to data processing. It's expensive but it packs a wallop. Take a look at a sample copy before you subscribe. 1-Year subscription $48.00.

Warren, Gorham & Lamont, 210 South Street, Boston, MA 02111.

ORGANIZATIONS

NATIONAL CONSORTIUM FOR COMPUTER - BASED MUSICAL INSTRUCTION

The NCCBMI provides a Forum for the exchange of ideas among developers and users of computer-based systems for musical instruction, establishes and maintains a library of music courseware, reduces redundant effort among courseware and hardware developers, and provides consultation for new users of computer-based musical instruction. It meets semi-annually, and held its last meeting Feb. 22-25 at the Hotel duPont, Wilmington, Delaware.

Contact Fred T. Hofstetter, NCCBMI, Music Department, University of Delaware, Newark, Delaware 19711.

INTERNATIONAL CONF. ON COMPUTER MUSIC

Annual meetings are held for members to present papers, attend discussions, give demonstrations and concerts of computer generated music. To date the meetings were at Michigan State University in 1974, at the University of Illinois in 1975, and at MIT in 1976.

Contact Barry Vercoe, ICCM, Room 26 — 313, MIT, Cambridge, MA 02139.

INVITE AN ACM SPEAKER TO YOUR HOME

The ACM audio library contains over 180 speakers on 49 cassettes from four ACM Annual Conferences, including the A.M. Turing Award Lectures, from 1972 through 1975. The cassettes are professionally edited and transcribed onto high-quality, low cost audio cassettes. Free catalog.

Information Cassettes, Dept. ACM, 645 N. Michigan Ave., Chicago, IL 60611. (312) 944-2120.

BASIC

The handle for the tool.

A microcomputer without software is a tool without a handle. PolyMorphic systems BASIC is the handle on the POLY 88; this provides the interface between user and computer. Our BASIC fits the POLY 88 like a finely balanced handle fits a quality tool. PolyMorphic Systems Basic is an extremely efficient way to program. It makes possible the immediate use of the POLY 88 for a wide range of engineering, scientific and general problem solving. In conjunction with either System 7 or System 16 PolyMorphic Systems BASIC is the reliable and ready to go microcomputer tool on the market. No more waiting for the long-promised software system. Among our best BASIC features: graphic plotting function, tape save and dump with named files, time function, and self-explanatory error messages. We believe the best tool — the POLY 88 — must have the best handle: PolyMorphic Systems BASIC.

Polymorphic Systems 11K BASIC

Size: 11K bytes.
Scientific Functions: Sine, cosine, log, exponential, square root, random number, x to the y power.
Formatted Output • Multi-line Function Definition • String Manipulation and String Functions • Real-Time Clock • Point-Plotting on Video Display • Arrays of up to 7 Dimensions • Cassette Save and Load of Named Programs • Multiple Statements per Line • Renumber • Memory Load and Store • 8080 Input and Output If Then Else
Commands: RUN, LIST, SCR, CLEAR, REN, CONT, LINE, NULL
Statements: LET, IF, THEN, ELSE, FOR, NEXT, GOTO, ON, EXIT, STOP, END, REM, READ, DATA, RESTORE, INPUT, GOSUB, RETURN, PRINT, FILL, OUT.
Built in Functions: FREE, ABS, SGN, INT, LEN, CHR$, VAL, STR$, ASC, SIN, COS, RND, LOG, TIME, WAIT, EXP, SORT, CALL, EXAM, INP, PLOT.

Systems Available. The POLY 88 is available in either the kit or assembled form. It is suggested that kits be attempted only by persons familiar with digital circuitry. The following is a list of the systems available.

System 1: is a kit and consists of the Poly 88 chassis, CPU and video circuit cards only. Requires keyboard and TV monitor for operation. **$595.**

System 2: Consists of System 1 with the addition of the cassette interface circuitry-requires cassette recorder. **$690.**

System 3: System 2 with 8K memory card, BASIC and assembler cassette tapes. **$990.**

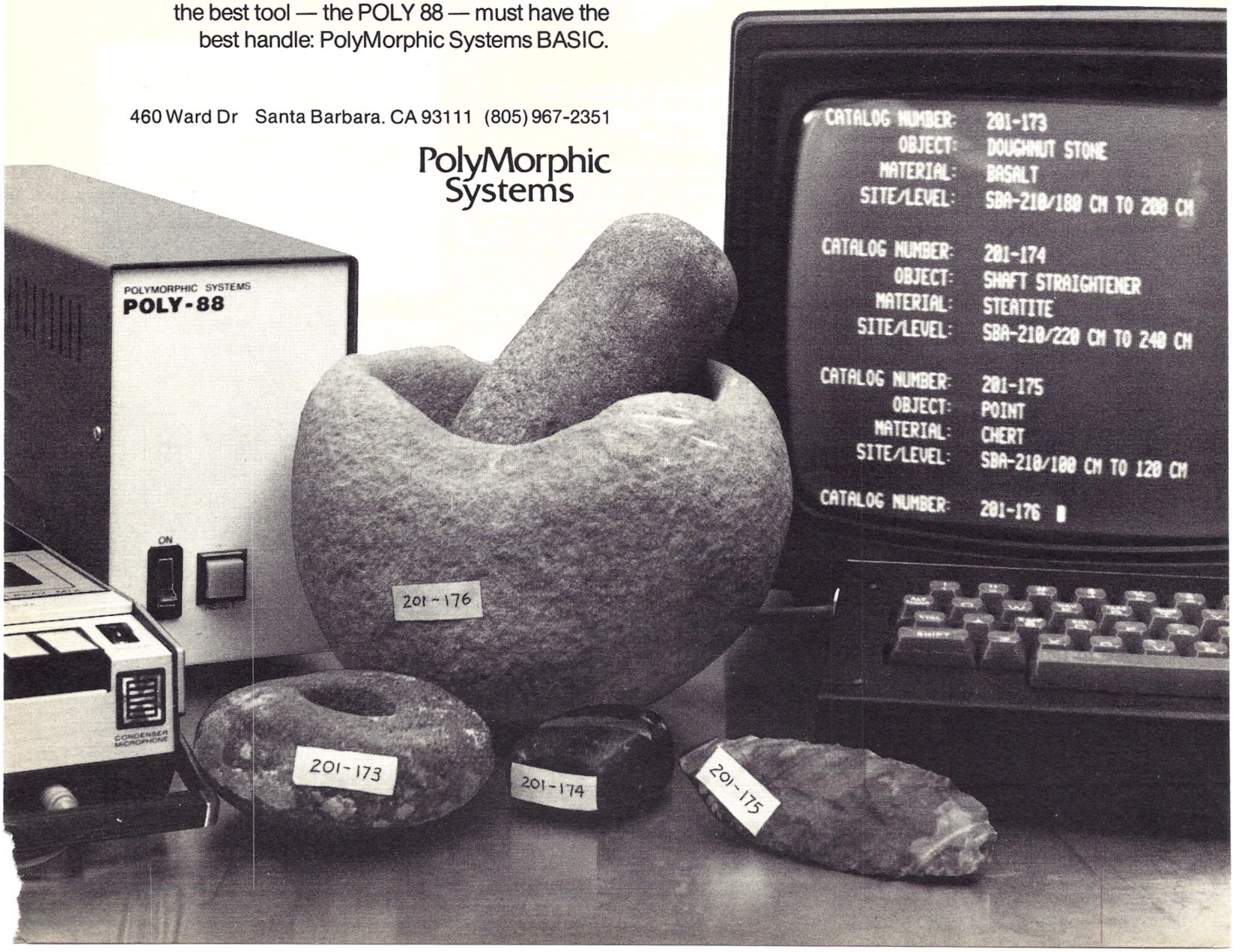
System 4: Also a kit, but containing in addition to System 3 a keyboard, cassette deck and TV monitor. **$1350.**

System 7: Consists of an assembled and tested POLY 88 with 8K of memory, keyboard, TV monitor, cassette recorder, 8K BASIC and Assembler cassette tapes. **$1750.**

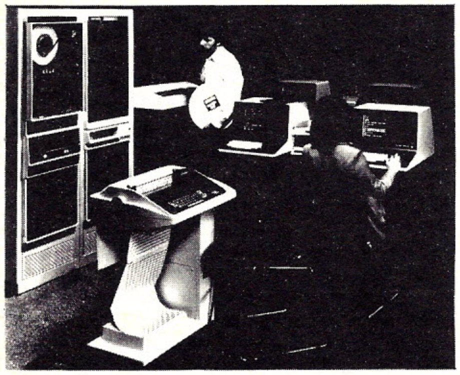
System 16: Consists of an assembled and tested POLY 88 with 16K of memory, keyboard, TV monitor, cassette recorder, 11K BASIC and Assembler cassette tapes. **$1995.**

460 Ward Dr Santa Barbara. CA 93111 (805) 967-2351

PolyMorphic Systems



COMPUTERS

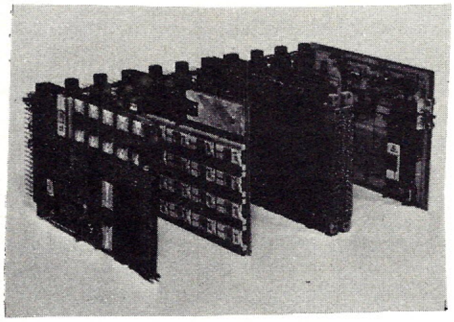


MULTITASK, REAL TIME MEGAMINI

The top of the line Nova 3/D computer uses 32K-word MOS memory modules and features a Memory Mapping and Protection Unit (MMPU) that allows concurrent batch and timesharing. The system supports Fortran IV with real-time extensions, multi-user extended Basic and Algol. This 16-bit "megamini" supports up to 128K words of memory and the complete line of Nova printers, terminals, discs, tape drives and other peripherals.

This system in medium or larger configurations should be of interest to schools for concurrent instructional timesharing and administrative (or instructional) batch. A medium-scale system with 48K memory, 10 megabyte disc and two hard copy terminals would cost around $37,000. A typical large-scale configuration with 96K memory, two 10 megabyte discs, 75 ips tape drive, 300 lpm printer, two HS hard copy terminals, and 4 CRT terminals would run about $87,000. Extensive literature available free.

Data General Corp., Route 9, Southboro, MA 01772. (617) 485-9100 X2556.

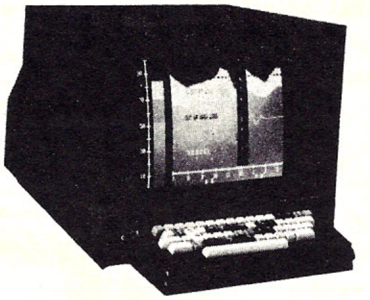


WINTEK REDUCES MICRO PRICE 50% TO $149, ANNOUNCES FANTOM-11

Wide customer acceptance of its WINCE MICRO MODULES has allowed WINTEK Corporation to halve the price on its single card microcomputer. The price was reduced from $298 to $149 for the minimum configuration WINCE CONTROL MODULE consisting of a 6800 MPU, clock and baud rate generator, 1K ROM with FANTOM-11, 128 byte RAM, and ACIA (UART serial I/O) or PIA (16 TTL lines parallel I/O). The price for the maximum configuration module consisting of 6800 MPU, clock and baud rate generator, 1K ROM with FANTOM-11, 512 byte RAM, ACIA (UART serial I/O) and 2 PIA's (32 TTL lines parallel I/O) was reduced from $398 to $199. All WINCE modules (control, RAM, ROM, EROM programmer, A/D, etc.) are on industry standard 4½"x6½" inch printed circuit boards.

FANTOM-11 is a new 1K monitor/debug program that allows single step execution of user programs, insertion and deletion of break points, and set up of interrupt vectors. It also allows the user to load memory, examine and/or change memory, print and/or punch memory, display MPU registers, go to user's programs, and reset.

WINTEX Corp., 902 N. 9th St., Lafayette, Ind. 47904. (317) 742-6802.



A COMPLETE 8-COLOR INTELLIGENT CRT TERMINAL KIT

Up until late last year, you may have seen an eye-catching ad for the Intecolor 8001 kit, "a complete 8-color intelligent CRT terminal kit," for $1395. Some who saw the ad considered the 8001 too expensive. Others, who read the specs a little more closely, felt the price was within reason. For $1395, you got an 8080 MPU, 25-line by 8-character display, 4k RAM, PROM firmware, 19-inch color CRT, RS-232 I/O, selectable speeds to 9600 baud, ASCII set, keyboard, and bell.

There's a new ad now, which says "It's an intelligent terminal for $1395. Or your personal computer for an additional $1295." The total price for the personal computer isn't given, perhaps on the theory that putting the $2690 amount in print might scare away even the aficionados. For the extra $1295 you get BASIC on ROM, additional 8k RAM workspace, insert/delete, background color, lower-case ASCII characters, roll, 48 lines instead of 25, 2X character height, "and a graphics mode with 160X192 elements."

Intelligent Systems was marketing the $1395 intelligent terminal kit to hobbyists, and an assembled unit to OEMs (original equipment manufacturers) and commercial users for $1995 ($1776 during a limited period, one to a customer). The assembled 8001 will still be available to OEMs, but the future of the $1395 kit is uncertain, and there is talk of phasing out both Intecolor kits. The hobby market has been just too difficult to support; there are too many phone calls, and many "screwed up" 8001 cards are sent in by hobbyists who can't get the terminal to work.

Starting last December, for $2995, a subsidiary company, Compucolor Corp., provides a similar terminal, assembled, also with BASIC and 19-inch screen, but sold only through computer stores. The Compucolor 8001 features an optional, separate "floppy tape memory," which turns out to be an 8-track audio cartridge, with a continuous loop, and single and dual drives. The shortest tape is 36 inches long, providing 16k bytes of storage, and a maximum retrieval time of 5 seconds. For more storage, use a longer tape. There may possibly be a Compucolor 8001 kit, at around $2495.

A cartridge library is in the works, to eventually include graphics, computations, check-book balancing, education instruction and tutoring, and computer games, including Star Trek.

For current information on these units, contact Intelligent Systems Corp., 5965 Peachtree Corners East, Norcross, GA 30071, (404) 449-5961.

TERMINALS

PORTABLE COMPUTER TERMINAL WITH ASCII AND APL

A new lightweight, dual code Execuport portable terminal that can be carried anywhere in an attache-type case is now available. The new Execuport 3000 Series terminals provide both ASCII or APL code operation and a graphic display of printed data based on their ability to print with a resolution of 240 (24 vertical by 10 horizontal) points per square inch. This unique graphic printing capability is made possible by the simultaneous movement of the print head and paper positioning.

Desk-top weight of the lightest model in the new series is only 22 lb. 8 oz. making it completely portable. A built-in acoustic coupler that accepts an ordinary telephone handset makes it possible for the terminal to communicate with a remote computer from any location with telephone service. It accepts 80 - or 136 - column paper rolls in 100 or 180 foot lengths.

Out of paper alarm, Bell decoder, 20 MIL TTY Loop, and many other options are available. The Model 3000 is completely

compatible with paper tape and magnetic tape memory units, most other peripheral equipment and all major minicomputers. The Execuport 3000 can be interfaced with a full array of peripherals by means of RS-232 connections at the rear of the unit. Price $3,500 and up; 7½% educational discount. Computer Transceiver Systems, Inc., East 66 Midland Avenue, Paramus, N.J. 07652 (201) 261-6800.

INTERACTIVE INTELLIGENT TERMINALS AND MORE

CDC recently announced five new entries in the interactive terminal arena. The 751-10 display terminal is a sophisticated unit controlled by an Intel 8080 microprocessor. It displays 128 ASCII characters (I didn't know there were that many) and 33 control codes. The keyboard detaches from the display to provide configuration flexibility. Half- or full-duplex operation from 110 to 9600 baud. The MPU permits a variety of transmission protocols. $3150 purchase, $93/month lease.

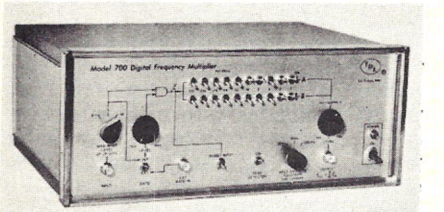
The 753-10 non-impact printer is a tabletop unit for use with the 751-10 display. The MPU controls the printer. Speed is 30 cps, 80-column lines. $2540 purchase, $74/month lease.

A higher speed (180 cps), multi-form (up to 5-part), any size (4-16.75") printer is now available. A full-line (132-character) buffer and a 1000-character buffer for shorter bursts of data are built-in. $4370 purchase, $126/month lease.

A single and dual cassette drive were also announced that operate at 7½ ips and store 288K and 576K characters respectively. $2200 and $2520 purchase, $70/$79 per month lease.

Control Data Corp., P.O. Box O, Minneapolis, MN.

PERIPHERALS



DIGITAL FREQUENCY MULTIPLIER

The TDL Electronics' 700 Digital Frequency Multiplier was originally developed to add polyphonic capability to monophonic music synthesizers but has numerous other uses. A rear panel connector provides a TTL interface so that the computer, not the front panel toggle switches, supplies the multiply and divide numbers.

The '700' multiplies the frequency of an input signal by the ratio A to B and provides fractional ratios. $769.

Ron Tipton, TDL Electronics, Route 7, Fayetteville, Arkansas, 72701. (501) 643-2191.

WRITE YOUR OWN SYMPHONY

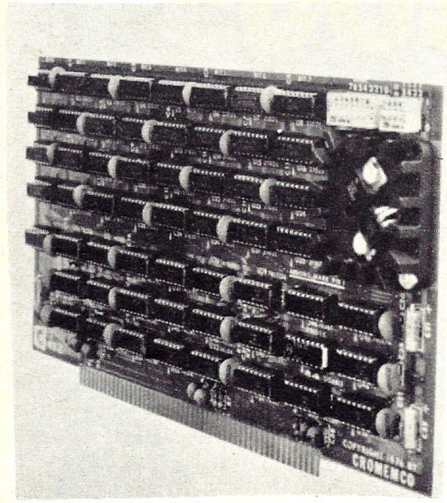
How would you like a machine into which you could tap out a short Bach Invention, and then play it back—Forwards or backwards? Or speeded up or slowed down? Or even "upside down," i.e., with the time intervals reversed? How about in a different key? Even if you're not a good pianist you can tap out a one-finger melody and add a generated rhythm accompaniment and, zap, you've got a finished composition. Or if you're more into special effects you can generate bomb blasts, ocean waves, sirens, or trains complete with whistles. Would you then like to see all your aural fireworks translated into visual patterns on a TV set? No problem — it's all part of the system.

Alfred Mayer, inventor of the Performer Synthesizer thinks the main value of this type of device is for teaching music. "Kids today don't want to spend hours learning. They want to make music right away and this lets them play around with real stuff right away. And learn!" He says that his device "reduces music to logic. It gets away from the fingers, the physical. And the physical is why more people don't play music."

The main elements of the system are the Ionic "Performer" synthesizer and modulator ($1295), Digionic sequencer, actually a computer in diguise ($999), and optional Ionicamera and TV display ($395). A wide range of other peripherals are also available. Catalog free.

Alfred Mayer, Ionic Industries Inc., 128 James St., Morristown, NJ 07960. (201) 539-1040.

MISC. HARDWARE



4K STATIC MEMORY

Cromemco's unique new RAM card is expandable to 8 banks of 64K each or a half megabyte of memory. The new memory operates at 4 MHz (to match the industry's fastest CPU card, the Cromemco Z-80 card). In addition, the new RAM card has the unique feature of *bank select*. This convenience for those who need more than 64K of memory makes it feasible to expand to *a half megabyte of memory*.

The fast 4 MHz speed of the new Cromemco 4KZ RAM card is achieved by using an address anticipation strategy. Addresses are applied to the memory chips *before* address information appears on the address bus. In this arrangement on-board address counters are incremented at the *end* of each machine cycle in preparation for the *subsequent* cycle. The result is that proven and reliable 21L02 low-power memory chips can be used at 4MHz. A wait state is automatically inserted only when two consecutive addresses are not sequential.

The RAM does, of course, also give ultra-reliable operation at 2 MHz clock rates with no wait states whatever. $195 Kit, $295 assembled.

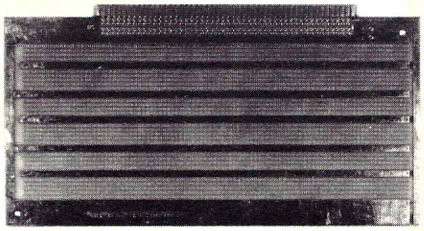
Joe McCrate, Cromemco, Inc., 2432 Charleston Road, Mountain View, Ca 94043, (415) 964-7400.

ANALYZER AIDS IN DIAGNOSING 6800 MICROPROCESSOR SYSTEMS

A new low-cost analyzer designed to develop and debug microcomputer systems built around the Motorola 6800 microprocessor is a product of AQ Systems. The instrument can display all address, data and status information and permits direct user interaction with memory and all registers including the program counter.

The Model AQ6800 Microprocessor System Analyzer is especially effective as a design aid in converting breadboard circuits to prototypes, and for the easy evaluation and fault analysis of microprocessor-based products in production. The addition of a buffered clip-on probe converts the analyzer into a portable production test or field maintenance instrument. The connection to the system being analyzed is easily accomplished with a buffer isolated probe terminated with a 40-pin clip that attaches directly to the microprocessor chip. $875.

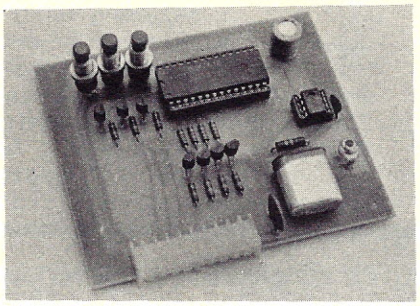
E&L Instruments, Inc., 61 First Street, Derby, Conn. 06418. (203) 735-8774.



PROTOTYPE BOARD

The Tarbell Electronics Model 1010 Prototype Board for the ALTAIR* and IMSAI accepts up to 33 14-pin IC's, or a mixture of 40-pin, 24-pin, 18-pin, 16-pin, and 14-pin IC's. It is mainly oriented toward soldering point to point, but wire-wrap may also be used. There are three rows for IC's. The IC's or sockets are inserted from the top, then wires are soldered to the adjacent tabs. The tabs are such that even with 40-pin IC's, there are still two holes left over on each pin for wires. A place for a 5-volt regulator is also provided. Edge pins are gold-plated. $28.

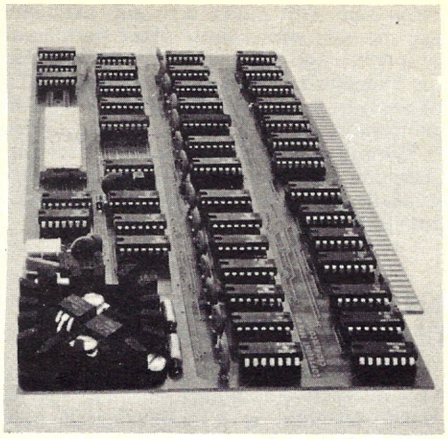
Donald E. Tarbell, Tarbell Electronics, 144 Miraleste Drive #106, Miraleste, CA 90732. (213) 538-4251.



CAN YOUR COMPUTER TELL TIME?

This hardware real time clock allows your computer to keep track of the time of day for control, timer, and game applications. This compact board contains a crystal oscillator which provides a 60 Hz. signal to a clock chip which, on command from your CPU, returns the digits representing the time of day in the form HHMMSS. Requires +5 and +12 volts. This versatile peripheral can be used with practically any microcomputer, even a 4040! $39.95 wired and tested plus 75¢ postage.

TED, P.O. Box 4122, Madison, WI 53711.



FAST CPU CARD

Cromemco has introduced an important new CPU card based on the new Z-80 microprocessor. The new card is by far the fastest known to be available. It uses a selected version of the Z-80, a version having a clock rate of 4 MHz (which is twice as fast as those using previous microprocessors).

Cromemco's new CPU card is designed as an easy way for the user to apply the Z-80 chip to his circuitry or system. For example, the card is plug-compatible with existing microcomputers. Further, it uses the industry-standard "S-100" computer bus which is supported by more than a dozen manufacturers.

The crystal-controlled 4 MHz clock rate of the new CPU card gives the user twice the throughput available with former cards but has also been designed to be compatible with 2 MHz systems and has a switch-selectable 2- or 4-MHz clock rate. It has the ability to jump to any 4K boundary in memory upon power turn-on. The new card has also been designed to operate with slower memory or I/O devices even when using the 4 MHz clock rate. This has been achieved by incorporating jumper-wire-selectable wait states in the card design.

Cromemco also has a broad line of peripherals that can be used with the new card. The peripherals include an 8K PROM card that has its own programmer, a BASIC firmware module, high-speed 4K and 16K RAMs, a fast 7-channel digital-to-analog I/O, a joystick console with speaker and amplifier, a color graphics interface, and others. It is supplied with a powerful Z-80 monitor, complete documentation, source code, and paper tape object code. A Z-80 assembler and BASIC interpreter are also available. $295 in kit form or $395 assembled. Delivery is 15-30 days.

Mr. Joe McCrate, Cromemco, 2432 Charleston Rd., Mountain View, Ca 94043, (415) 964-7400.

SOFTWARE

PROGRAM LIBRARIES FOR TI SR-52 CALCULATOR

Two free program libraries with a suggested retail of $59.90 will be available to purchasers of the Texas Instruments SR-52 magnetic card programmable calculator between January 20 — March 31, 1977.

Purchasers who take advantage of the offer will have their choice of two prerecorded libraries covering mathematics, statistics, finance or electrical engineering. Those libraries have a suggested retail of $29.95 each.

During the first quarter of 1977, TI will re-institute a program initiated last fall offering a $10 rebate to purchasers of the SR-56 key programmable calculator.

Texas Instruments, Calculator Products Div., P.O. Box 5012, Dallas, TX 75222. (214) 238-2011.

8080 EDITOR, ASSEMBLER, DEBUGGER

Tychon offers 3 software packages for 8080 users. The Symbolic Editor (nickname TED) is used to create and modify symbolic (source) statements and to transfer them to and from paper tape using a Teletype. TED requires 1600 memory locations and 120 words of scratchpad. Additional memory is used for user text.

The Assembler (TAS) converts source programs produced by TED to a binary tape and Teletype listing. TAS can assemble from either paper tape or the text buffer in memory. TAS is a 2-pass assembler that requires 2.6K memory for source code, data, and scratchpad. To use TED and TAS your system should have at least 4K memory and a TTY with paper tape reader and punch.

The Debugger (D-BUG) allows a program to be executed starting at any address, inserts breakpoints at which time the value of all registers and pointers are typed out, allows altering of one or more memory locations, and permits single-step operation.

TED and TAS tapes $25, listings $40. D-BUG tape $10, listing $40. Documentation packets alone (for everything) $5. Information free.

Tychon Inc., P.O. Box 242, Blacksburg, VA 24060. (703) 951-9030.

MICROCHESS

Microchess is a chess playing program written for the Kim-1 6502 microprocessor system. All moves are entered and displayed via the Kim Keyboard and LED display. The program can be adjusted to one of three levels of play, requiring 3, 10, or 100 seconds for each computer move. The level of play is below that of a very good player but offers challenging moves. The system is expandable and fully documented. It has a Player's Manual, complete annotated source listing, and Program Documentation describing the strategic algorithms. Instructions are provided for modification, expansion or system conversion. $10.00.

MICROCHESS, 1612-43 Thorncliffe Park Drive, Toronto, Ontario, M4H 1J4, Canada.

8080 BASIC INTERPRETER

In case you're not familiar with interpreters, they differ from compilers in that they directly execute program statements, in contrast to the separate machine code program generated by a compiler. While often slower than a compiled program, it is generally easier to debug a program with an interpreter since one has only to evaluate source statements.

Called BASIC ETC, this interpreter uses 8K bytes of memory plus 1K scratchpad. Features include 255-character strings, N-dimensional arrays, variable precision arithmetic, assembly language subroutines, direct memory and I/O addressing, 27 error codes, character and line erasing, editing, subroutine nesting, 31 commands and statements, 8 functions plus user defined functions, null control (0-25 seconds), and formatted output statements. Altogether, a very rich Basic. Available on paper tape or audio cassette complete with 32-page manual. $25.

Binary Systems, Inc., 634 S. Central Expressway, Richardson, TX 75080. (214) 231-1096.

2 NEW MODULES FOR NCR SCHOLARS SYSTEM

NCR has released the remaining two modules for its SCHOLARS data base system for student records management. The purpose of SCHOLARS (School Automated Records System) is to automate all aspects of records management, including attendance, scheduling, grade reporting, test scoring, and academic history, and thereby make it more efficient. The new modules are for test history evaluation and academic history record keeping. SCHOLARS is now in use in over 50 sites ranging from elementary schools to four year colleges and universities.

NCR Corporation, Main and K Streets, Dayton, Ohio 45479

OPUS/ONE

A.S.I. has announced a significant price reduction for their OPUS/ONE compiler/interpreter; the INTEL 8080-compatible language has been reduced from $300.00 to $99.00. A complete User's Manual is available for a nominal $5.00. The reason for the price reduction was given as the tremendous interest generated by previous announcements. The new prices have been made *retroactive*, and all of those who purchased OPUS/ONE prior to the price reduction have been reimbursed for the difference.

The current version has drivers for all Altair I/O interface boards, though drivers may be inserted by the user for other I/O devices. OPUS/ONE is fast and efficient in memory utilization. It incorporates the strong points of several large-system languages such as ALGOL and FORTRAN, yet maintains the commands, statements and simplicity of BASIC. Some highlights of the language are: *arithmetic precision* ... up to 126 digits; *strings* ... automatically converted to numbers during numerical operations, with a length of up to 127 characters; *GOTO, GOSUB* ... label references are literal strings within the program; *variables* ... virtually unrestricted in character length and can represent a number, string or matrix; *matrices* ... up to 255 dimensions with either number or string elements; *formatting* ... I/O print format statement has right- and left-justification, carriage return/line feed controls within the parameter list; *block structure* ... similar to ALGOL's BEGIN-END features, brackets delimit blocks of program code. Delivery: 30 days.

Administrative Systems, Inc., 222 Milwaukee, Suite 102, Denver, Colorado 80206. (303) 321-2473.

STAR TREK

ADVANCED STAR TREK IN PL/I: PRICE HIKE

The price of cards for the complete game in 1600 statements offered in the Sept/Oct issue has gone up from $10 to $18. Still cheap for 2000 cards plus postage.

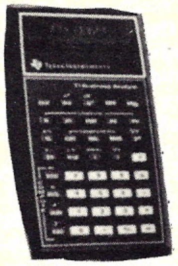
John R. Bane, P.O. Box 3125, University Station, Clemson, SC 29631.

DICE STAR TREK

Play the Mayfield Star Trek + bits of Leedom/*Creative Computing* Super Star Trek at home, with or without a pocket calculator. Dice simulate random numbers. You need a chess board and pieces, one die and masses of paper and fun-time. Issued at virtually no profit, produced by stencil duplicator, runs to 50 pages. Price $2 or £1 sterling, whichever preferred.

Marion Stubbs, 176 Midanbury Lane, Southampton SO2 4GX, England.

CALCULATORS



BUSINESS ANALYST

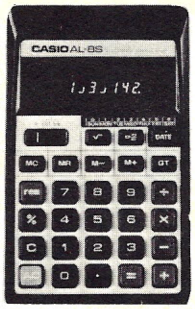
The new Texas Instruments "Business Analyst" is a powerful yet economical tool that provides fast answers to problems involving the time value of money.

Built-in linear regression and trend line analysis capabilities allow direct interpretation of statistical information. Additional points can also be predicted for forecasting purposes, compound interest, annuities, investment yields and many other complex personal or business matters that need quick solution with reference to complicated tables can be done with the TI Business Analyst.

The compact, six-ounce calculator is easy to use. It has complex mathematical algorithms pre-programmed. All a user must do is enter the known variables (number of periods, interest rate, present value, future value or payment). Then the calculator computes the unknown factor. To examine the effect of changing any variable, the user enters that new variable and recomputes.

Built for trouble-free operation, with a single integrated circuit at its heart, warranted for one year from purchase against defective materials and workmanship, the Business Analyst sells for $49.95.

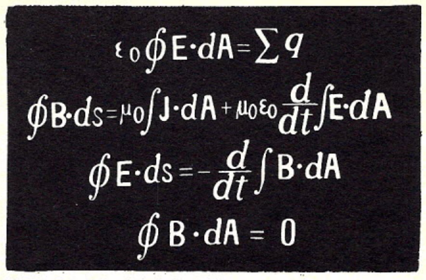
Texas Instruments Inc., P.O. Box 5012, Dallas, TX 75222.



NEW CALCULATOR FEATURES FRACTIONS AND DATES

The *AL-8S* by CASIO is the world's first fraction/date calculator. At the touch of a button, the user can add, subtract, multiply and divide numbers expressed in fractions. The answer is also displayed in fractions. Other highlights include: Computes day of the week for any date between 1901 and 1999; Computes day intervals between any two dates in the 20th century; Independent memory with four key memory M+, M-, MR and MC) and grand total memory with "GT" key; Remainder calculation in division; Square root at the touch of a button; Per cent key for mark-ups and discounts; and 8-digit capacity with large, bright green display. $24.95.

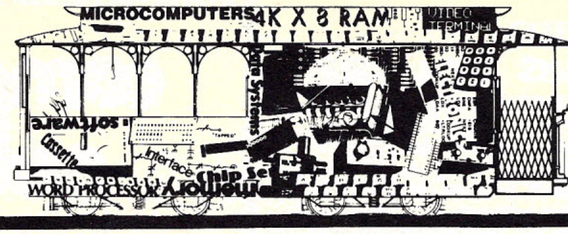
MISCELLANEOUS



WILD T-SHIRTS!

How would you like a T-shirt with the quadratic equation, Maxwell's equations, or LaGrange's equations? No, you're not into equations — well then how about the periodic table, the circulatory system, psychology maze, or the molecular structure diagram of chlorophyll. Or do you prefer something with more culture like a novel composed of 16 first and last lines from classic and well-known novels or perhaps the opening bars from Bach's Invention 1. And for the electronics freak a transistor flip-flop schematic or a logic diagram of a feedback circuit. Perhaps you'd like to do your own thing with a blank punch card (IBM beige on a navy shirt) and an indelible pen to "punch" the holes. All shirts in great color combinations! $4.50 per shirt, 50¢ postage per order. Catalog free.

Outer Products, Dept. C, Box 88, Lafayette Hill, PA 19444.



THE FIRST WEST COAST COMPUTER FAIRE

A Conference & Exposition on Personal & Home Computers

San Francisco Bay Area – Where It All Started – Has Its First Home Computing Convention

7,000 to 10,000 People
100 Conference Presentations
200 Commercial & Homebrew Exhibits
Two Banquets with Outstanding Speakers
Special-Interest Social Centers
Publication of *Proceedings* Being Planned

In Northern California's Largest Convention Facility, San Francisco's Civic Auditorium

CONFERENCE SECTIONS ON HOME COMPUTING

Being Planned

- Computer Graphics on Home Computers
- Computer-Driven & Computer-Assisted Music Systems
- Speech Synthesis Using Home Computers
- Computers & Amateur Radio
- Computer Games: Alphanumeric & Graphic
- Personal Computers for the Physically Handicapped
- Computers & Systems for Small Businesses
- Tutorials for Hardware Novices & Software Novices
- Software Design for Personal Computers
- Microprogrammable Microprocessors for Hobbyists
- Optical Scanning for Inexpensive Program & Data Input
- Floppy Disc Systems for Home Computers
- Hardware & Software Standards for Personal Systems
- Seminars for Club Leaders, Editors, Organizers, etc.
- Personal Computers in Education (associated with a University of California short-course)

–– AND MUCH MORE ––

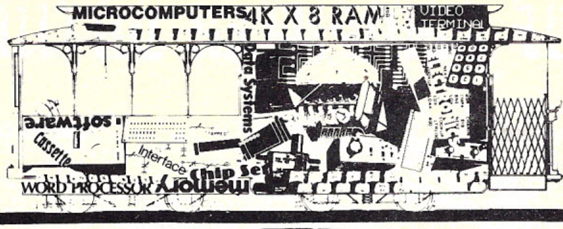
PRESENT-WORLD & FUTURE-WORLD BANQUET SPEAKERS

Banquets to be Held in San Francisco's St. Francis Hotel

Fascinating Speakers will Discuss the Past, Present, & Future

- Frederik Pohl, Science Fiction Writer; Lecturer to NASA, American Astronautical Society, New York Academy of Sciences, World Future Society, etc.
  Robots You Can Make for Fun & Profit
- John Whitney, Pioneer Computer Film Maker, Computer graphics experimenter under grants from NEA, Guggenheim Foundation, & IBM
  Digital Pyrotechnics: The Computer in Visual Arts
- Henry Tropp, Mathematician, and Principal Investigator for History of Computers Project jointly sponsored by the Smithsonian Institution & AFIPS
  The 1940's: The FIRST Personal Computing Era
- Ted Nelson, Author & Fantast, Director of the Xanadu Electronic Literary Network, Lecturer at Swarthmore College
  Those Unforgettable Next Two Years

april 15-17, 1977 · san francisco



THE FIRST WEST COAST COMPUTER FAIRE

A Conference & Exposition on Personal & Home Computers

FIND OUT ALL THE DETAILS –

GET YOUR **FREE** *COPY* OF THE

SILICON GULCH GAZETTE

- Details of the programs & speakers
- Information about the banquets' arrangements
- Accommodations information, & registration forms for the St Francis Hotel (Faire Headquarters Hotel), & the other three Computer Faire hotels
- Pre-registration details for the Faire (reduced admission for those who pre-register)
- Details of the *Proceedings* expected to be published
- Articles about the homebrew exhibits
- Weather information, & tourist trivia regarding San Francisco
- Listings of the commercial exhibitors
- *AND*, just to make it interesting:
  - "Hot news", & raging rumors from "Silicon Valley" (the San Francisco Bay Area)
  - Product announcements, equipment descriptions, hardware & software news, etc.

CO-SPONSORS INCLUDE AMATEUR, PROFESSIONAL, & EDUCATIONAL GROUPS

- The two largest amateur computer groups:
  Homebrew Computer Club
  Southern California Computer Society (SCCS)
- Both Bay Area Chapters of the Association for Computing Machinery (ACM)
  San Francisco Peninsula Chapter
  Golden Gate Chapter
- IEEE Computer Society's Santa Clara Valley Chapter
- California Mathematics Council
- Stanford University's Electrical Engineering Dept.
- University of California's Lawrence Hall of Science
- People's Computer Company (PCC)
- Community Computer Center
- Bay Area Microcomputer Users Group (BAMUG)
- Professional & Technical Consultants Assn (PATCA)
- Amateur Research Center

april 15-17, 1977 · san francisco

Cut out, or photocopy; then fold, and tape in position.

☐ Send a *free* copy of *Silicon Gulch Gazette.*

☐ Send program & accommodations information, & Pre-registration Form (those who pre-register may take advantage of a reduced admission fee).

☐ Send speakers' instructions.

☐ Send homebrew exhibitors' instructions.

☐ Send commercial exhibitors' prospectus.

☐ Send details on the expected Computer Faire *Proceedings.*

PLEASE PRINT OR TYPE

NAME ____________________

ADDRESS ____________________

CITY ________ STATE ________ ZIP ________

77-1-12b
CC

COMPUTER FAIRE † (415) 851-7664

BUSINESS REPLY MAIL
No Postage Necessary If Mailed In U.S.A.

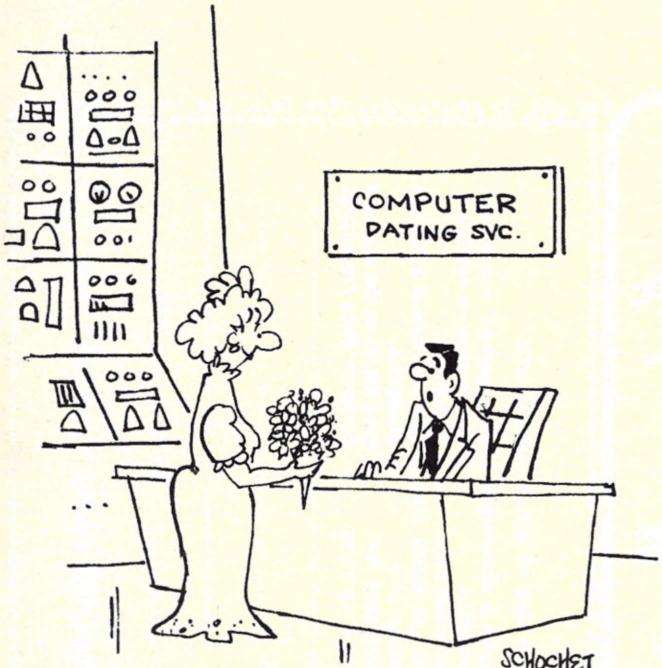
FIRST CLASS
Permit No. 91
Palo Alto, CA

POSTAGE WILL BE PAID BY

THE FIRST WEST COAST COMPUTER FAIRE

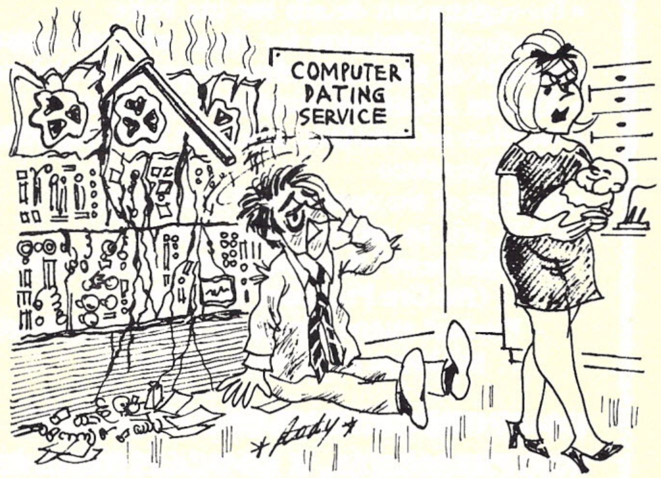
BOX 1579
PALO ALTO CA 94302

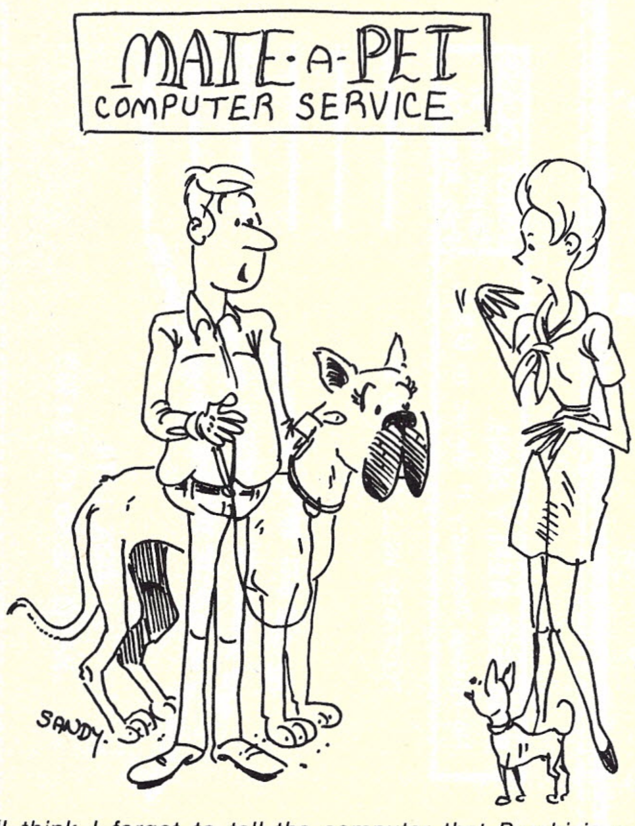
THE LIGHTER SIDE OF COMPUTER DATING



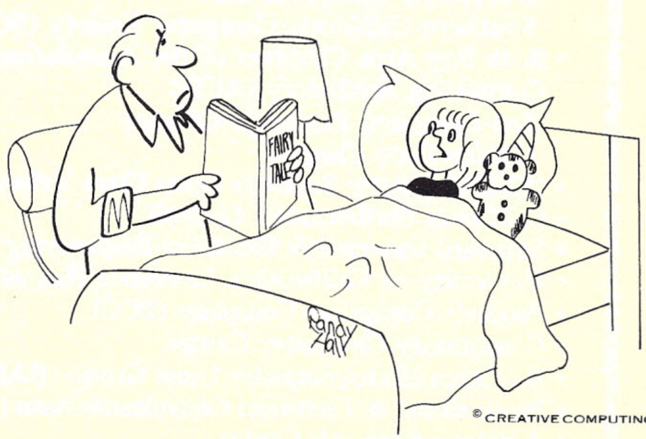
"Let's not rush things now Miss Jablonski."

"To think I needed a computer to find YOU!"





"I think I forgot to tell the computer that Bambi is a Great Dane."



"Cinderella and the Prince could have made it a whole lot easier on themselves if they'd used computer dating."

ATTENTION READERS

Creative Computing is seeking articles, programs, graphics, games reviews and other material for future issues. We are especially looking for "how-to" material. Contributions should be typed, double-space (3000 words maximum). Include a self-addressed stamped envelope if you want an acknowledgement.

We pay for material from professional writers at rates between 1¢ and 3¢ per word. Please note on your submission whether you are submitting on a paid or unpaid basis (the probability of acceptance is higher for unpaid material). We normally do not pay for material from educators or students.

Upcoming issues will focus on a wide variety of topics including the following:

Computer kits
Arts and humanities
Music
TTY and plotter graphics
Health care, medicine
Psychiatric treatment
Public access (libraries, museums, storefronts)
Retail computer stores
Networks
Computer conferencing
Education
Computer games that learn
Robots and cyborgs
Future computers

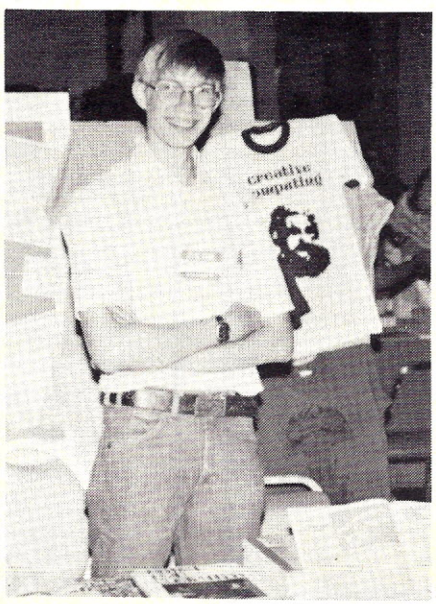
In which Publisher Dave Ahl playfully describes the genesis and on-going development of a home computer system and Contributing Editor Steve North describes succinctly, but sometimes irreverently, some of the components in this and in his system. Beware! This could happen to you!

Saga of a System

My story starts off back in June of 1975. Actually it goes back a lot further than that. Since the late 60's, I've always had a time-sharing terminal at home. But in the last few years my kids began to use the terminal for CAI, mainly drill and practice in mathematics and, of course, they like to play games. Obviously, this affected my telephone bill as well as sending the bill for time-sharing computer time through the roof. So, having seen some of the ads and articles about these new little do-it-yourself computers I decided that I would get one. In early '75 the only one that I could get any significant information on was the ALTAIR 8800 so I decided to place an order for one. I ordered it around the middle of June '75, and got it about 60 days later in early August.

Then I decided that if I built it myself and wrote up my experiences I would probably be a little bit biased. Since I certainly wanted to describe what it was like to build a computer system on the pages of *Creative Computing* I decided the best thing to do would be to find some unbiased outsider, who had experience in both writing and computer systems. So I called my good friend Steve Gray, and said, "Hey Steve, I got this neat Altair 8800 coming in a couple of weeks. How would you like to build it?" Steve, not to pass up a golden opportunity like that, said "Sure, be happy to, I've got a couple of free evenings, some week-ends. I'll build it for you." Well it probably took a little bit longer to assemble than either Steve or I had originally anticipated and compounding the task was the fact that I had ordered a fairly complete system around the basic Altair 8800: 8K of dynamic memory in the form of two 4K boards, a serial input/output interface for a terminal and another input/output interface for an audio cassette recorder as well as the CPU board itself. So that was our basic system. It took Steve many many hours to put the whole thing together. His experiences are well documented in the article that appeared in the Jan/Feb 1976 issue of *Creative Computing,* "Building an Altair 8800".

A couple of months elapsed in early 1976 when Steve's job seemed to be taking precedence over his hobby; just couldn't spend an awful lot of time on the Altair and things were sort of in limbo. Then in March I took the opportunity to go out to Albuquerque to the MITS Altair convention and I saw in person for the first time a TV Dazzler. In fact, I saw several of these rather impressive little devices. I decided that was just the thing I needed for my Altair so Harry Garland, President of Cromemco, and I had a conversation which culminated with me ordering a TV Dazzler along with a Bytesaver board to store programs for both the Dazzler and the Altair. They arrived just a few short weeks later and once again I entrusted them to my friend Steve Gray (who was probably getting a little tired of the whole thing by now). I asked Steve if he would build the Dazzler which he did (the Bytesaver I had gotten assembled for some strange reason).



Contributing Editor Steve North

We then decided it was high time to get the system to a point where it was usable. After all, I still had the time-sharing terminal at home and I was still running up my telephone bill and bills with GE time-sharing. So I said, "Hey, let's shoot for the National Computer Conference in June. Let's get the Altair-TV Dazzler set-up running with a little kaleidoscope program or some other catchy demo and show it at our NCC booth."

Well, about that time we discovered, as many other buyers of TV Dazzlers have no doubt discovered, that the TV Dazzler does not work with dynamic memory — it requires static memory. That posed a minor problem, so I took advantage of another acquaintance, Tom Kirk of the N.J. Amateur Computer Group, who happened to have a spare 4K static memory board. I borrowed it thinking, "I'll run with this one until I can get my own." Tom probably has had some second thoughts about loaning me that memory since it was not until November that he finally got it back. Which reminds me of the story: when a visitor asked Mark Twain, "Why do you have so many books stacked on the floor and all over every room of your house?", he replied "Because my friends won't lend me their bookshelves."

In any event, we didn't quite get the Altair, Dazzler, and other bits and pieces running in time for the National Computer Conference. The main reason was because of one of those "other bits and pieces," in particular a little device called a Pixieverter. Now a Pixieverter is a small enough device; it fits in the palm of your hand and only costs $8.50 from ATV research. But unfortunately Steve's Pixieverter was acting up and at that point I had not gotten one of my own.

SAGA — Continued . . .

What a Pixieverter does is convert low frequency signals put out by the Dazzler, TV camera, or other similar device to a higher frequency that can be fed into the normal antenna terminals on a TV set so that you don't need a special TV monitor or don't have to modify the TV set on the inside.

Back to NCC. What to do? The obvious solution: call upon another friend. This time Bob Radcliffe from Hoboken Computer Works. I explained, "Bob, I've got this exhibit at the National Computer Conference and I'm kind of committed to showing a jazzy hobbyist computer system (along with a Tektronix 4051 on the other side of the booth to show that we're still legitimately involved with assembled products for the education market). How about loaning me a set-up?" Well, he very kindly did that and we had a very nice exhibit with a Hoboken Computer Works machine (an IMSAI 8080 in disguise) with a TV Dazzler running kaliedoscope. It turned out to be a real big attention-getter in our booth.

However, back to our system. What was happening? Well, in fact not very much was happening and I decided, "Gee, it's been a while. I'd like to see what my computer looks like. Maybe it's about time that I take a look at it." I figured that I had a little spare time on my hands and maybe it's about time to take soldering pencil to hand. So I got the computer back from Steve along with all the other little bits and pieces. Lo and behold I discovered I really didn't have nearly as much time to work on it as I thought so I got in touch with one of our prolific contributing editors, Steve North, who had just graduated from high school a few miles away from me. I said, "Steve, how would you like a nice little summer project to get this computer system running. There really shouldn't be much to it, it's all assembled, it's all checked out, it just doesn't quite run yet. The first step is to get and build a Pixieverter of our own." No problem. Steve built it and it seemed to work

Dazzle Your TV With a TV Dazzler

If your computer is dull and boring, and only blinks its lights and types number and letters, perhaps a TV Dazzler is for you! Physically, a TV Dazzler is two boards which plug into your Altair or IMSAI, connected to each other with a ribbon cable, and which drive a TV monitor or modified TV. The TV Dazzler generates (to simplify a bit) a color picture of your computer's memory, similar to the VDM-1. But instead of generating characters, the TV Dazzler makes little colored squares appear on your TV screen. You can use it to create all kinds of interesting color graphics.

The Dazzler uses high-speed direct memory access (DMA) to read the memory of the host computer and translates the information into a TV picture while at the same time the computer is executing a program. There are two output ports on the Dazzler, and one input port. One output port tells the Dazzler whether or not to display a picture, and if so, where the starting address of the picture is in the computer's memory. The other output port is used to tell the Dazzler what resolution to use, whether to generate B&W or color; and for high resolution, what colors to use (blue, green, and red are available).

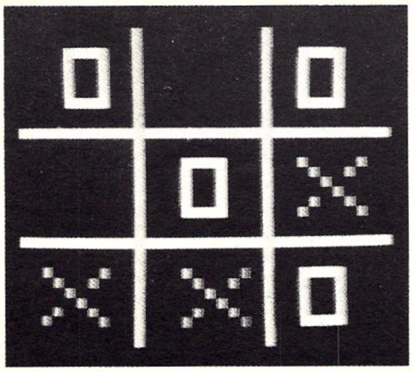
The Dazzler can generate a 32x32 element picture, which uses up 512 bytes of memory, or a 64x64 element picture, which soaks up 2K of memory. The Dazzler's memory is not on-board, so you'll have to reserve some static memory in your system for it.

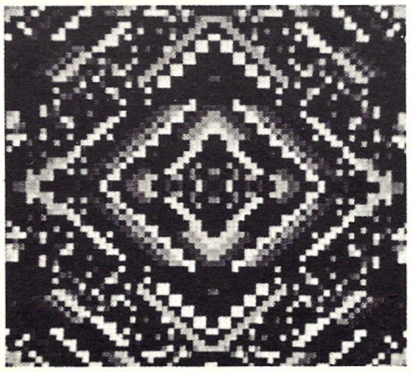
Only two bits of the input port are used. Bit 7 goes low during odd lines, and high during even lines on the TV. Bit 6 goes low for 4 msec to indicate the end of a frame.

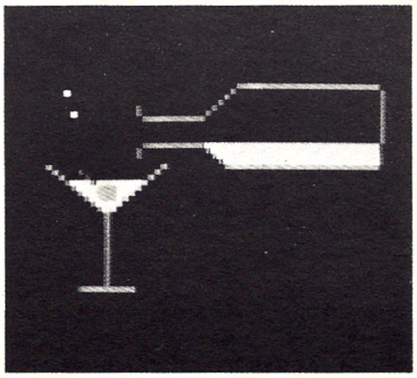
In the normal (low) resolution mode each 4-bit nybble of memory represents one square on the screen. The individual bits specify which colors and what intensity. In the high resolution mode, things get a bit trickier. Each byte represents which squares are on or off in a group of 8 squares. To decide what colors and intensity to use, the Dazzler looks at the appropriate output port. To get full color in the high resolution mode, you have to interweave frames of different colors. But enough of that here; a very comprehensive set of instructions and technical description comes with the Dazzler.

The Dazzler board itself seems of better construction than most PC boards; it uses sockets throughout and goes together very nicely as long as you don't have too heavy a hand with your soldering pencil. A simple test program is included for alignment of the Dazzler; however the alignment is somewhat tricky since all the adjustments seem interrelated! Extensive troubleshooting hints are included in the manual; at this point we haven't needed them—reliability has been excellent.

There is already a fairly extensive Dazzler software library. Kaleidescope is a Dazzler simulation of something you can get in a 5&10 for 39¢, and LIFE is a version of the popular computer game for Dazzler graphics. Dazzlemation is used to generate animated Dazzler pictures (such as an endlessly pouring Martini), and Dazzlewriter is used to write alphanumerics on the Dazzler. This program seems a little cumbersome to use, because the Dazzler doesn't have the resolution for getting many characters on a line. If you get an A/D with a joystick or two (such as Cromemco's unit) you can run things like Dazzle-Doodle, which lets you draw color pictures on the Dazzler; a chase game; and a space war game (real space war, not Star Trek) to be reviewed in an upcoming issue of *Creative.*







Processor Technology VDM-1

In any usable computer system, you need some way of getting people-oriented information out of the system. Quite often this takes the form of a hard copy device or a CRT terminal. If you have an 8080 based system with a 100 pin bus, you have a third option — a VDM. The VDM board plugs directly into the system bus, and a cable runs off to a TV monitor. Although the VDM does display characters on a TV screen, it isn't a CRT in the usual sense. In other words, you don't send out characters to it one at a time, as you do with most computer terminals. Rather, the VDM is a visible page of memory. Each byte shows up on the TV screen as a corresponding ASCII character, both upper and lower case. Since ASCII uses only seven bits (and there are eight bits in a byte), the eighth bit is used to turn the cursor on or off at a particular location.

Since you can't just send characters out to the VDM, a special routine is required to write characters out to the VDM's memory (which is on-board incidentally — the memory for the picture doesn't eat into what you already have). This routine to simulate a terminal takes up about 1/2K of memory, but it lets you do things you can't do with many CRTs. While characters are being written on the VDM, you can alter the speed of the display by typing a number from 1 to 9. 1 is ultra-slow (about 1.5 char/sec), while 9 is equivalent to about 2000 lines/minute! You can stop the VDM from displaying more characters by hitting the space bar. Subsequent depressions of the space bar cause characters to be displayed one at a time. Typing another number causes the display to continue at the desired rate. Typing control-A will turn the cursor on or off, depending on its previous state. The display scrolls, which is much more readable than starting over at the top of the screen when it's filled.

With VDM also does several handy things with its on-board hardware. An output port is used to tell the VDM how many lines to blank at the top of the screen, and the address of the on-card RAM where the display starts. There are also some DIP-switches which permit you to select: 1) black on white, or white on black picture; 2) whether the cursors blink or not; 3) whether the control characters are visible or not; and 4) whether the screen is blanked to the end of the line on a carriage return, and to the end of the page on a vertical tab. An input port on the VDM has one bit which goes low if the output port wasn't written to in the past .25 seconds.

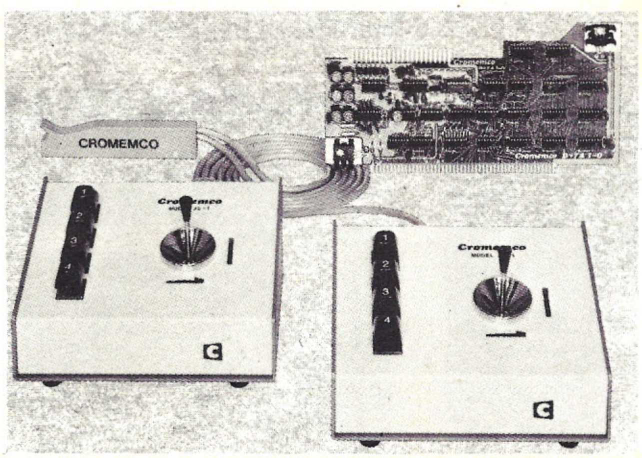
Unfortunately, there is no way to connect a keyboard to the computer through the VDM; you'll have to obtain a separate interface for that. Still the VDM is useful as the only human-readable output device in a system, and perhaps even more valuable to someone who already has a hard copy device (save time, paper, and wear-and-tear on the ears)!

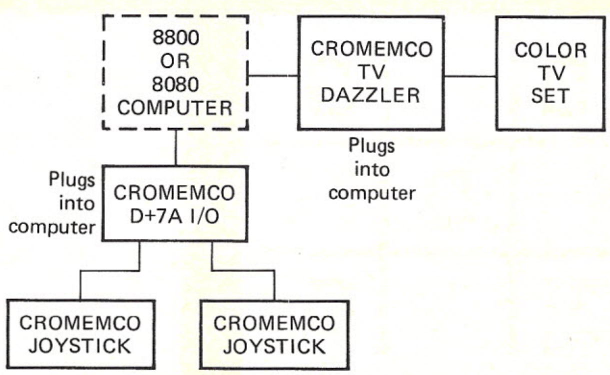
Software with the VDM includes the standard terminal replacement software, and patches for MITS BASIC. PTCo 5K BASIC and the soon to be released 8K BASIC have built-in VDM drivers. Raise a sense switch, and you have output to the standard terminal — lower it and the output goes to the VDM. There are also some VDM games we haven't had a chance to try out yet, including a realtime Star Trek game. Since all the characters on the VDM are directly accessable to the computer, it has the capability for some interesting graphics.

Cromemco A/D and Joysticks

Once you have a device like a TV Dazzler in your system, you may get tired of watching it play Kaleidescope all by itself for a few hundred hours. You'll probably want some way of playing a game *with* the Dazzler. Of course, you could use a regular keyboard for input, but joysticks lend themselves more readily to graphics, and can be used by children who haven't learned to read yet, or by people who are put off by computers. Cromemco's D+7A is a seven channel A/D; a set of matching joystick consoles is also available (you can buy one or both). The A/D itself is bipolar- that is, it accepts voltage as input in the range of -2.56 to +2.56 volts and digitizes it with sign in 40 mv increments. The A/D board also has a parallel digital I/O port.

The joystick consoles feature a joystick (of course), a speaker, and four pushbutton switches which can be read by the computer. Since two joysticks use only 4 channels, you still have 3 channels left over for other miscellaneous things. Let your computer listen and talk to the real world! Construction of both the D+7A board and joysticks are outstanding as we've come to expect from Cromemco. While an A/D setup with joysticks isn't the most essential thing you can plug in your computer, it does make your computer more fun.





SAGA — Continued . . .

fine. As a matter of fact, after a little poking around, fine tuning, putting in a capacitor here and there, and replacing one of the integrated circuit bus drivers on the CPU board, finally, the computer got working along with the TV Dazzler. Great! Story finished. Well, not quite.

Towards the end of the summer I made another mistake by going to the Personal Computing Fair in Atlantic City. I said "mistake," however, it was not a mistake going to the show. What happened was that Harry Garland from Cromemco brought along his TV Dazzler along with a couple of joysticks and the mistake was that I brought along my kids and showed it to them. After that, the two of them — that is, the Dazzler/joysticks set-up and my kids — turned out to be inseparable. All three kids just went bananas over this system. So once again I decided this is something worth having (actually, my kids decided) and once again I placed an order for an analog to digital interface (this time assembled) along with two joysticks (yes, not one but two).

They arrived in short order and I plugged them into the system. Recall, the Dazzler worked fine — it ran kaliedoscope just beautifully so one would expect that a preassembled A/D converter with two joysticks would also run fine and would produce little traces across the screen of the TV set when you moved the joystick appropriately. But instead of little one dot traces when I turned the thing on I got great big gashes of color diagonally across the whole face of the tube. It took two minute movements of the joystick to cover the entire face of the tube with color. This was not quite what I had in mind, thinking back to this little electronic etch-a-sketch which my kids were playing with in Atlantic City. What was wrong? Well, I didn't know what was wrong so I called up Cromemco and said, "This thing gives me these gashes of color—what's wrong?" "Well sir, we can't really tell long distance but we can make a pretty good guess that you've got some ripple in the power supply so why don't you put a big capacitor across the power supply and see whether that works?" I said, thinking back to my days in radio/tv repair, "Does that mean a 35 microfarad capacitor or a 50 microfarad?" The telephone laughed at me and finally the engineer said, "No, it means something like, 10,000 microfarads."

Well, I rummaged around in my box of old electronic parts and found that I had a 350 microfarad unit but nothing like 10,000 microfarads. This created a small problem because 10,000 microfarads aren't available in your neighborhood Radio Shack store and anyway I wasn't absolutely sure that that was the nature of the problem.

Anyway, right around this time also I had taken my system up to the Science Fiction Conference at Great Gorge Playboy Club where it proved to be the delight of all of the people with hangovers in the morning who couldn't quite maneuver a bunny or themselves to their room the night before. A fair number of the people who looked at the TV Dazzler kind of went away with dazzles in their eyes. However, the point is that we experienced some minor problems up there due, apparently, to the fluctuating or lower line voltage. So I said to myself, "Gee, I go to a lot of different trade shows and conventions. I should really have some kind of constant voltage power supply or an auto transformer or something so that this system can be easily transportable." So I decided that perhaps a Parasitic Engineering power supply was the answer and placed an order for one. Back now to the problem of gashes of color across the face of the tube. Well, looking at what could have caused it, first of all the voltage used in they joystick mechanism is the +16 to -16 and it's regulated down to 12 volts in the joystick but, of course, for the regulators to work you'd better start off with a little bit more than 12 volts. I put a volt meter across the terminals and while just the CPU was in I got a nice 16-1/2 volts from the 16 volt power supply. Add a few more boards and it dropped down to 15-1/2 volts; more boards, 15 volts; then 14-1/2 volts; finally plugged in the two joysticks and all of a sudden the -16 volts had dropped down to -12.3 volts. Since -12.3 probably wasn't being regulated to -12 very well by the Zener diodes it was pretty apparent that that was the cause of the problem. The +16 and the two 8 volt sources were still doing fine. I determined then that it would probably take a very simple fix of a 12.6 volt filament transformer to replace the 11 volt transformer (T3) in the Altair to fix the problem along with perhaps a larger capacitor than the 500 microfarad one in the -16 volt power supply, say something like 2000 or 3000 microfarads seemed about right. Just when I had reasoned all of this out, along comes a newsletter from the Denver Amateur Computer Group with precisely the same power supply fix suggested. That is, replacing transformer T3 with a 12.6 volt filament transformer etc., etc. I could have put in this fix for about $2.69 for the filament transformer plus $1.25 for the capacitor from one of the surplus outlets but since I'd already blown $75.00 on the Parasitic

Cromemco Bytesaver

It seems to be becoming more popular to keep some programs on PROM, save thereby save yourself the agony of toggling in a bootstrap loader every time you bring up your system. The Bytesaver is a PROM board with a built-in programmer. It has room for 8K of (or 8) 2708 PROMs. Unfortunately, we can't even find 2708 PROMs in the back of *Popular Electronics,* but Cromemco sells them preprogrammed for $50. Does that give you a hint? The Bytesaver comes with a program called Bytemover on a PROM. Bytemover is used to transfer programs from RAM to ROM (for programming) and from ROM to RAM (so they can be run). By setting the sense switches, you tell Bytemover exactly what transfer to make. Obviously this gives you a very fast way to load programs into RAM (but not many of them). To program a PROM, you have to flip a switch on the Bytesaver board, and then run the Bytemover program with the proper switch settings.

We really question the worth of the Bytesaver. It does program Erasable PROMS — which are much more flexible than the non-erasable type. But these EPROMS do not appear to be widely available. Since it takes big bucks to buy PROMS for the Bytesaver, you might consider a smaller PROM board, containing at least a bootstap loader. On the other hand for "permanent" software (BASIC, a sort routine, favorite games, etc.) the Bytesaver might be just the thing.

SAGA — Continued . . .

Engineering power supply I decided I might as well go ahead with that and leave the Radio Shack/surplus vendor power supply fixes to other people who aren't quite so foolish with their hard earned money as I am. Actually I shouldn't really say that because my justification for getting the Parasitic Engineering power supply was more because it provides a good source of constant voltage under varying input conditions rather than just to fix up the gash in the TV Dazzler/joystick combination.

Meanwhile about the same time, Steve North mentioned that the system seemed to be having some CPU problems. Steve had put a capacitor across two of the terminals in the CPU to get rid of some noise in the Dazzler image. It also seemed not to be performing adds up in the high registers correctly. So I wrote MITS about the problem and returned the CPU board. Well I got back a nice little note from MITS saying that the CPU checked out just fine and when it is used with *MITS* products everything would be ok but that they could not guarantee when it was not used with *MITS* products. They also performed the clock modification at the same time which was very nice of them. Unfortunately, when I ordered the Parasitic Engineering power supply I also ordered their clock fix kit which uses a somewhat different strategy of fixing the clock than the MITS kit.

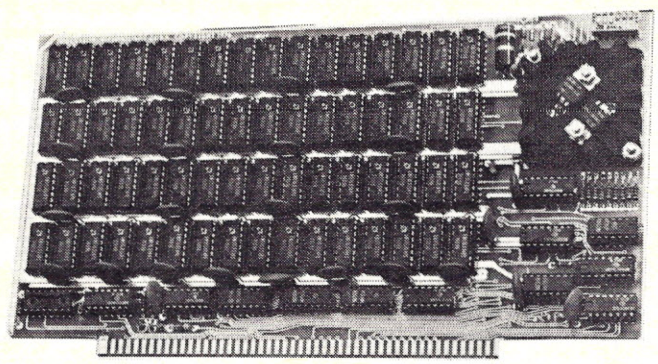
It's pretty obvious that the direction that I was going with the Altair was to make it an analog device driving a graphics terminal of some sort, say a color tv set along with they joysticks. About that time I said to myself, "what happened to the original purpose of getting a Basic speaking computer to replace the time-share terminal link up?" It wasn't hard to deduce that somehow my goals had been subverted along the way but that my original goal was still worthwhile and therefore I really should have dedicated a computer to speak Basic to my kids. Having read so many good things about other microcomputer chips such as the Motorola 6800, Zilog Z80, Intersil 6100 and so on I decided I really should have one of them. Further, having heard a lot of good things about the Southwest Technical Products 6800 kit I decided to get one. But that is a subject for another story some time.

Back to the Altair. Somewhere along the way I had to cure the problem of static vs dynamic memory. It was eventually cured by getting an 8K static memory from Processor Technology along with a video display.

Processor Technology 8K Static RAM

One static memory is as good as another, right? Well, when it comes to actually storing data, yes, provided that it is fast enough for the CPU, and doesn't get flaky, or overload your power supply (everyone's claiming low-power drain these days, but some are lower than others). The PTCo 8K does offer some extra frills though. All the ICs are socketed, a worthwhile feature. It has a dip-switch which permits you to select the starting board address in 1K increments. The 8KRA also has a plug for connection to a battery, so when you turn your computer off, the data on the board is retained. To be frank, we don't know anyone who actually uses this feature, but it seems to be the "in" thing in memories nowadays. At least it's there if you want it . . .

Assembling the 8KRA is a simple operation. You just solder IC sockets until you begin to see IC sockets even when your eyes are closed! As with all closely packed boards, you have to be careful not to make any shorts between traces while soldering.



The 8KRA also comes with two test programs. One is for a system with just a front panel and an 8KRA, while the other is for a system with a terminal. The simple program tests all the words in the first 8K memory (except for the portion of memory the program itself is in, of course), and when it finds a bad word, quits, and stuffs information pertinent to the error (such as the address of the bad word, what the program tried to put there, and what it actually read back, etc.) in the first few bytes of memory. But, surprise! The program will always tell you your memory is bad, because the first address it checks in 8192 decimal, not 8191. While there are 8192 words of memory on an 8K RAM, they start at location zero, not one. Someone goofed. Additionally, the explanation of what gets stored in the first few bytes of memory if there's an error was a little muddled. The other test program prints out a neat little picture of the ICs on the board, with X's for bad chips, and G's for good ones. Impressive! Show off to your family and friends! Despite the criticism, remember that unless you only know how to run memory checks, you're buying some well designed hardware, and not software with some minor ambiguities.

I'm not quite sure why I got the VDM other than the fact that it looked that the Altair was shaping up into something that would use a television set generally for output and it looked like the Processor Technology video display module had a lot of interesting performance characteristics.

So at this point my system looked something like this: Altair 8800 CPU; 8K Processor Technology static memory; a Bytesaver board, TV Dazzler, 7 channel analog/digital and two joysticks from Cromemco; an audio cassette interface and terminal input/output boards from MITS which at this point weren't doing much of anything, a Video Display Module from Processor Technology which also at this point was not doing much of anything, and a 4K dynamic memory from MITS (which at some place along the line I decided to upgrade with the new synchronous kit so it's 4K of updated dynamic memory which is doing absolutely nothing. The other 4K of memory was given to Tom Kirk in payment for the loan of his static memory over a rather prolonged period of time.) The output of the TV Dazzler and Video Display Module both go into the Pixieverter which is a rather inconvenient operation to switch from one to the other. I intend to try to make that a switched feature in the near future. The whole thing drives a small GE color TV set. Also I intend to hook up the cassette interface board to an audio cassette recorder and use BASIC on the system. I also intend to hook up the serial interface I/O board to some terminal as yet undetermined. (One problem of working for Digital Equipment and now for AT&T — at DEC I got hooked on a DECwriter which is a little bit out of my price league to put out for my own computer system. Now working for AT&T I'm hooked on the KSR 43 terminal which also is a little bit out of my price league. I guess I'll have to settle for something else, but at this point I haven't decided quite what.

Just a general note about trends in plug-in components of various manufacturers. Since the Altair originally came out, an entire industry has sprung up to make Altair compatible boards and even CPU's. I can't help but be very impressed with the quality of these boards that are available from these other manufacturers and now from MITS itself, particularly compared to the original boards that came with that first Altair. The Cromemco boards are absolutely beautiful, they're double thickness, all gold plated contacts. The assembly instruc-

Processor Technology 3P+S

The 3P+S is a multipurpose I/O board for use with both serial and parallel peripherals. Here's what you get:

1. Two parallel ports. These are standard TTL level ports, with data strobes (input) and acknowledge flags (output). You might use a parallel port for interfacing a keyboard or optical papertape reader.

2. One serial port. This port can be used to send serial data on either an EIA RS-232 channel (which modems and CRTs generally speak) or on a 20 ma current loop (which Teletypes usually speak). There are lots of jumpers to select options on the serial port — jumpers for UART control (word length, parity, etc.), jumpers to select baud rate, a jumper to enable or disable the current loop, and jumpers to determine which (if any) of the EIA channels you send and receive on.

3. A control port. A variety of flags can be jumpered to the control word for input to the CPU. These include flags from the parallel ports and, UART flags (such as Transmitter Buffer Empty, Received Data Available, and the error flags).

When your computer is running, and wants to input or output data, it can look at the appropriate bit on the control word and determine if a particular device has data to be read in, or is ready to accept more data. The output from the control word can be jumpered in similar ways. Some bits can be used for software control of the baud rate and UART (so that, to change from one terminal to another, you'd change the program in the computer, not the jumpers on the board. Someone we know is using a 3P+S in his Altair to run a totally programmable modem!) You can also jumper bits of the control word, both in and out, to EIA channels you're not using for serial data, which is useful if your peripheral requires signals besides "Send," "Receive," and "Ground." There's even a peripheral control driver on the 3P+S.

In all fairness, the 3P+S should be called a "2P+S+C" because it only has two parallel ports. The board seems reasonably well designed, and should be all most home computer owners need in the way of I/O interfacing, except for a mass storage interface (cassette or floppy disc). The only thing we could fault the 3P+S on is Processor Tech's documentation. For instance, the manual never explains in one place what all the little jumpers really do. The manual does explain how to connect the 3P+S to some common peripherals, but they even slip up in the instructions for creating an RS-232 interface, by neglecting to tell you to jumper the TBE and RDA outputs of the UART to the control word inputs. Test programs for the board are included in the manual, but the test program for the serial port seems excessively long to toggle in manually. It might be a lot simpler to write your own test program, and if you have difficulties try the one provided, since it does have some diagnostic features. Despite these minor inconveniences, the 3P+S is a good value, and most hobbyists would find it more useful than either a serial or parallel interface alone.

Footnote: After finishing this article, we spent torturous hours trying to get the 3P+S to talk to a working TTY. The problem turned out to be documentation. Pins J and 8 on a connector are switched in a diagram. Although the manual has been recently corrected, ours wasn't, hence 4 wasted hours.

SAGA — Continued . . .

tions with the Cromemco peripherals were outstanding in their clarity and both peripherals ran without any trouble whatsoever. I have almost the same comment for the Processor Technology boards — they are very nice boards — gold plated contacts, heavy weight board and high quality throughout. Both Cromemco and Processor Tech use sockets throughout on their boards which I personally think is very worthwhile, particularly if you, like I, have to replace an IC. In one case I had to replace the bus driver, in another the clock chip, and in still another, I had to replace a whole board full of IC's when I updated the MITS memory from the original to synchronous memory. It is just not a fun thing unsoldering IC's. Southwest Technical Products make a very convincing argument for the fact that the IC sockets are just something else to go wrong with the system — bad contacts and so on — and have convinced most of their kit builders that these sockets are not necessary. They are probably correct, but on the other hand I would probably still lean toward sockets in the event that changes might be desirable at some later date. The Parasitic Engineering components can't be directly compared to peripherals on a board since, of course, transformers, the clock IC and other parts for the Parasitic kits don't use any boards. However, the quality of the components was very good throughout. (One minor problem with the Parasitic power supply was that it assumed that the capacitor C14 in the Altair power supply was a 2000 microfarad unit or above — maybe they're thinking of late model Altairs — because mine was a 500 microfarad capacitor.)

All in all, it has been quite an experience — three people have been directly involved — myself, Steve Gray, Steve North, and many more involved on the periphery including some very, very helpful people at MITS and Cromemco who quite willingly spent time with me on the phone when I had some problems and tried to answer my questions and, in general, were very helpful. That's a very impressive part of this whole industry. Recalling my days at Digital Equipment it was very much harder then, and probably still is, on the customers of minicomputers and larger mainframes. If a customer had a technical question and was trying to modify the equipment the first thing he'd hear is "Don't." The second would be, "It can't be done." Compounding it all

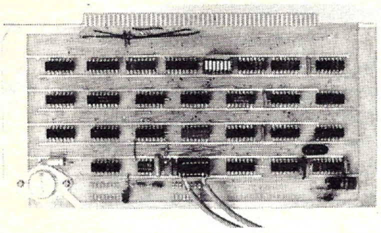
Tarbell Cassette Interface

For some time now, a cassette tape standard known as the Kansas City/Byte standard, has been in use in amateur computing. While this standard can be used with cheap recording equipment, it does have the disadvantage of being slow. With the Kansas City standard of 30 bytes/second, it takes 4 minutes to load 8K of memory. That's a long time to wait every time you bring up BASIC or some other software of that size!

The Tarbell cassette interface is also compatible with cheap recording equipment, but it can read and write data at 187 bytes/second. That means you can load 8K of memory in 43 seconds! Furthermore, if you're the adventurous type, you can try to modify the interface to run a 540 bytes/second, although you might find your bargain basement cassette recorder isn't quite up to the job. Tarbell is still recommending exchange of data at the 187 bytes/second speed.

The kit itself is fairly simple to assemble, it's just a PC board with 30 or so ICs, some discrete components, and a few jumper wires. The cables that are used to connect the interface to a recorder are also included. To adjust the interface, you just plug it into the computer, hook up the cassette recorder, turn everything on, and play a test tape consisting of nothing but sync bytes (E6 hex). You then twiddle a trimpot until a LED on the board comes on, indicating that sync bytes are being received. That's it.

The manual that comes with the kit includes instructions, recommendations on what kind of recorder to use, operating instructions, bootstrap and output programs with checksum, an example of how to use the interface under control of a BASIC program, instructions about how to modify the interface for computer-controlled start-stop, and patches for CLOAD and CSAVE in BASIC. Tarbell also markets two software packages: a chess program we weren't able to review, and a modification of Processor Tech's Software Package #1, an assembler/text editor/monitor. Tarbell's modifications permit you to write both source and object code out to the interface, and then read the code back in or check what you wrote for validity (so you won't save a big program and discover that the recording was bad and you've lost the file).



A tape of the software is $5 and a manual with a source listing is $5.

Obviously having a cassette interface that runs that fast and isn't expensive (depending on your point of view) is nice, but there are a few problems. First of all, there have been some complaints of problems with getting the interface to work. You're not really taking a risk, though, since Tarbell will repair the board free of charge, or refund your money, within 90 days. Another thing — you can't just write characters or code out to tape, and then read them back in. Whenever you write out a block of data, you have to first send out a start byte, then a sync byte (E6 hex), and then all the data. You don't have to bother with these control bytes when reading data back in — just reset the interface by sending a 10 hex out to the control port, and start reading. Also, since the interface is a synchronous device, with data being sent out in blocks, you can't do too much processing in between writing characters out- under 5300 instructions. Under most circumstances that wouldn't be a problem anyway; and if you do have to do a lot of processing, as in BASIC, you could always put all the data you want to record in a buffer and when you're done processing, write all the data out to the cassette at once.

The last problem is that there really is no cassette tape standard, with people using the MITS cassette interface, *Byte* standard, and Tarbell. But most software is available on Intel format papertape. If you have a friend with an 8080 based system and a papertape reader, you could always try to get him to copy whatever software you buy to a Tarbell cassette tape. Actually, the computer store I frequent (Computer Mart of NJ) is willing to sell you software on one of the usual mediums, and then copy it onto your own cassette.

SAGA — Continued . . .

was the fact that he probably couldn't reach the right person to talk to anyway. About the best you could do generally was to talk with a sympathetic field service technician who may not always be as knowledgeable as the customer himself. At DEC, we certainly had a number of customers that fell into that category. The general feeling was, "Oh, you'll void the guarantee," or "You'll screw up the system somehow and we can't be responsible and stand behind it if you do that." Well that may be quite true for the mini and mainframe manufacturers. But I think it's heartening that people working for the manufacturers of the microcomputer kits and peripherals and virtually everyone that I've come in contact with in this fledgling hobbyist industry is openly helpful. Although you occasionally get notes back as I did from MITS saying that it works fine with MITS products and so on, for the most part people are thoughtful and recognize that you're not going to use all the products from just one manufacturer in assembling a system — that's just not the nature of the hobbyist computer user. (I use the term "hobbyist" here very loosely because I'm including anyone who is building a microprocessor unit whether he's building it for a small business, for a school or just for fun at home).

If there's one area that the hobbyist computer manufacturers fall a bit short in it would have to be the area of documentation. The supporting documentation and checkout programs are, in many cases, very weak. In fact there is nothing with the Altair that I would put in the category of a diagnostic program; for example, something that checks every single bit of memory, turning it on and off, and then giving a check sum or diagnostic at the end. At Digital Equipment, field service technicians and software support people were armed with a case full of diagnostic and checkout programs for every single component in the system. This is severely lacking in many (most) of the hobbyist computer kits or peripherals I've seen to date. Consequently at this point there's a great necessity for clubs — the Amateur Computer Group of New Jersey of which I'm a member — or any one of the other several hundred computer clubs across the country where hobbyists can learn from the experience of one another what works, what doesn't work, what's likely to go wrong. Good old-fashioned trouble shooting techniques are also very practical. Fortunately a computer is a very logical device by nature and, in fact, is fairly easy to diagnose by simply tracing back through the circuitry and identifying what component or components are likely to be at the cause. (That's how I identified the bus driver IC that was malfunctioning in the Altair. It was simply a matter of going from one schematic diagram to another and tracing all of the components that had some influence on the signal, then trying combinations and seeing which things worked and which things didn't. Finally it narrowed down to just one component. Of course, it was not the entire bus driver that had burned up but just one of the gates).

This computer system which seems never to be quite finished is probably quite typical of most hobbyist/school microcomputer kits. Speaking for myself (and not necessarily for Steve Gray or Steve North, who actually had the burden of construction on their shoulders) it's been a real ball. And in the future, even though the self-contained black boxes are on the way, I look for increased fun as well as challenges from my system(s). ■

Parasitic Engineering

Parasitic Engineering sells two fix-kits designed to alleviate bugs in the original Altair 8800. One of these is a power supply kit, which uses a constant voltage transformer and high-current rectifiers. Unlike a linear transformer, a constant-voltage transformer, with the help of a special winding and a capacitor, maintains a constant voltage at the secondaries while the primary voltage can range from 90 VAC to as high as 140 VAC. To install a Parasitic power supply, you have to remove the back panel, and disconnect the power supply wiring. After replacing the original Altair P/S transformers with the big fat C-V transformer and capacitor, you have to do some minor surgery on the power supply board to install the high current bridge rectifiers. All that remains then is to reconnect all the power supply wiring. We found it necessary to install a Parasitic Power Supply in our Altair when we found that it didn't have enough muscle on the -16v bus with a full card cage, particularly with dips in the line voltage. The Parasitic Power Supply is rated at 12 amps at 8 volts, and 2 amps at ±16v. Not quite an IMSAI power supply, but adequate! (Note that the writer enjoys taking jabs at the Publisher's Altair.)

Parasitic also sells an Altair clock fix kit, which promises to do everything but debug your programs for $15. Since some early Altairs seem susceptible to clock sync problems, you might find it worthwhile. De-installation of the existing clock IC on the CPU board, installation of a 94618 in its place, as well as some cutting of old traces and replacement of a handful of resistors and capacitors takes an hour or less to do.

What we really want to know is, why did they name their company "Parasitic Engineering?"

Addresses

Here are the names and addresses of the manufacturers mentioned in the above article.

MITS, Inc.
2450 Alamo S.E.
Albuquerque, NM 87106
(505) 243-7821

Cromemco
2432 Charleston Road
Mountain View, CA 94043
(415) 964-7400

Processor Technology Corp.
6200 Hollis Street
Emeryville, CA 94608
(415) 652-8080

Tarbell Electronics
20620 South Leapwood Ave., Suite P.
Carson, CA 90746
(213) 538-4251

Parasitic Engineering
P.O. Box 6314
Albany, CA 94706

ATV Research
13th and Broadway
Dakota City, NB 68731
(402) 987-3771

Computer Mart of New Jersey
501 Route 27
Iselin, NJ 08830
(201) 283-0600

Greatest Computer Shows Ever!

Personal Computing magazine is proud to announce that it is sponsoring the first series of regional Personal Computing Shows.

Beginning with the *Western Personal Computing Show* in Los Angeles, and followed by the *Eastern Personal Computing Show* in Philadelphia and the *New England Personal Computing Show* in Boston, Personal Computing magazine intends to make everyone aware of low-cost computing.

Other shows are now being planned for the South, Southwest, Canada, and Europe!

Already, invitations have been sent to all the manufacturers in the personal computing field, computer stores, computer clubs and well-known computer experts.

Special areas of the exhibition halls will be set aside for Personal Computing in Education, in the Home, in HAM Radio, and in Small Businesses. These are all first for a computer show.

Seminars and special presentations include: Computer Synthesized Music, HAM Applications, Trends in Microcomputers, Mass Storage Systems, Lemonade Computer Service Companies, The Kitchen Computer, Computers on the Farm, The Small Business System, Software for Fun and Practical Applications, Computer Club Organization, Standards for the Hobbyists, Computer Art, The House Robot, Computer Crime, Software Protection and Future Computing.

In addition, *special tutorial workshops* will cover all aspects of computer hardware, programming in both machine language and higher-level language and applications. Workshops are designed for both beginners and advanced students in the art of personal computing.

We anticipate 150 different exhibits and crowds of up to 10,000 people at each of these shows. Arrangements for the shows are being handled by a professional management company to ensure that everything runs smoothly.

Cost of Registration:

At the door:

$10 per show (two days)

$ 6 per One Day Pass

Special Pre-Registration Rates:

$ 7.50 per show (two days)

$ 4.00 per One Day Pass

Note: Show tickets and one day passes entitle you to attend all seminars, workshops, exhibits and other events.

Register Now and Save!

Yes, I would like to take advantage of your special, pre-registration rates. I plan to attend the following regional Personal Computing Show(s):

☐ Los Angeles	☐ Philadelphia	☐ Boston
☐ Show (two days)	☐ Show (two days)	☐ Show (two days)
☐ One Day Pass Only	☐ One day pass	☐ One day pass

Enclosed is a check for ______________________

Name ______________________

Address ______________________

City ______________ State & Zip ______________

Send to: Personal Computing, Conference & Exposition Management Co., Box 844, Greenwich, CT 06830.

Could this be the answer to the need for a generalized form to present programs?

ALGORITHMIC BASIC

by Tom Allen
P.O. Box 81
Stevensville, MI 49127

Ed. Note: Recently I received a long letter from Tom Allen, most of which is reproduced below. If readers have reactions to the ideas put forth by Tom, please write him directly or send your thoughts to me at Creative Computing — DHA.

I read the letters to the editor and the "On Computer Languages" article in the Sept.-Oct. issue with a great deal of interest. It occurs to me that the question of the "proper" language for the *Creative Computing* context may be an example of confusing the medium and the message. After all, a particular program, written in a particular programming language, is only an implementation of a much more general solution algorithm. No well-designed algorithm depends on a specific language of implementation for its success. Moreover, it seems to me that a clearly-stated algorithm would be useful to a much wider group of potential users than would an implementation in some specific language. For instance, the primary language of the realm in which I currently reside is COBOL, in an RJE environment. So, in order to gain access to any of the programs you list in *Creative,* I have to extract the algorithm from the BASIC program, clean it up, and then implement it in COBOL. That's a'lotta work!

By way of translating these thoughts into something constructive, I have a suggestion to offer. Perhaps *Creative* could include a regular monthly column featuring the algorithm underlying some program appearing elsewhere in the issue or a program of wide-spread interest. Since I've spent a little time thinking about this idea, let me share some of those thoughts with you. First, it's a half-baked idea; the part that's still raw is the question of what language or meta-language should (or could) be used to state the algorithm. Tho there is an historic precedent for the use of ALGOL, my personal opinion is that this is a case of the cure being worse than the disease. And I don't think the graphic flowcharting language is the answer because of problems in trying to shoehorn a flowchart into an 8½x11 format for printing. It seems to me, however, that the so-called pseudo-language which the structured programming boys have been fooling around with lately might just be the answer. Using this language invites the additional use of the top-down, structured approach to the design of the algorithm as an additional plus. Anyway, I think that the problem of a language for the statement of algorithms can be dealt with reasonably well.

Getting algorithms out in the open so everybody can see them can produce another benefit as well. When the entire readership of *Creative* can see the algorithm, it then becomes possible for everybody to contribute to the refinement and generalization of an algorithm. As an example of this, I've been fooling around with a COBOL version of the Lunar Lander. Once I got the basic algorithm for the Lander out in the open, I began to see all sorts of extensions and generalizations that could be gotten almost for free with a suitable statement of a top-down, structured version of the algorithm. It also allowed me to "segregate" the physics behind the simulation in a very comfortable and understandable way.

The one facet of this idea which bothers me somewhat is a concern about the problems which the "typical" *Creative* reader/programmer might encounter in converting an algorithm into the program. The syntactical structures available in such languages as COBOL, ALGOL, and PL-I lend themselves easily to the implementation of a top-down, structured algorithm. This is less true of FORTRAN, and even less so of BASIC. Moreover, if one is using BASIC in a minicomputer environment (where else?), it is often necessary to be "tricky," and parsimonious in one's coding techniques so as not to exceed core limitations. Sometimes it's damn difficult to be both elegant and stingy!

So there's the idea, for what it's worth. I'd be most interested in your reactions.

One last item: do you know of anybody whose got an algorithm of a good random number generator or an implementation thereof in a higher-level language — like COBOL, for instance. I'd sure rather get one ready-made than have to take the time to learn the theory necessary to develop one myself!

Ed. Note: On receipt of the letter above, I replied to Tom encouraging him to pursue the idea further and possibly work out an example or two. A few weeks later, I received his reply.

I've been doing considerable thinking since I wrote you last about the problem of expressing algorithms in a way that would be appropriate for *Creative Computing.* I've worked out an approach that feels pretty good to me; I'd be very interested in your reactions.

My approach grew out of the thought that, even tho there are segments of the *Creative Computing* readership that do not wish to write programs in BASIC, it is none-the-less probably true that close to 100% of the readers can read and understand BASIC. Therefore, why not use a kind of "pseudo-BASIC" as an algorithm language?

After some experimentation, I've roughed out some of the properties and guidelines for a "psuedo-BASIC" algorithm language. First, a major difference: *no* line numbers in pseudo-BASIC. Instead, I've gone to a section/paragraph structure very much like COBOL. This allows one to segment the logic blocks of an algorithm quite nicely and to name them in a very meaningful way. The scope of sections or paragraphs is just as in COBOL: the scope extends only to the next occurring section or paragraph name — no nesting of scopes. Statements which would normally take line numbers in their syntex (such as GO TO, IF . . . THEN . . . , and GOSUB) now take paragraph names.

Secondly, I've confined my use of BASIC to a set of "care" statements—no extended or generalized statements. My feeling here is that clarity and immediate understandability are more important than economy of expression.

I've also tried, insofar as it is reasonable, to adhere to the spirit of structured programming: no superfluous GOTOs or multiple exits from a sub-process. Finally, I've tried to make consistant use of indenting in the writing of an algorithm, to exhibit the nesting and sub-process structure of the algorithm. I've also gotten in the habit of using a non-standard variable "REPLY" for YES/NO type responses

from a sub-process or the TTY—like a kind of special flip/flop arrangement.

Rather than try to talk this thing to death, I've whipped up a few examples of the application of these ideas to some programs from previous issues of *Creative Computing.*

This second example, an algorithm for NOT ONE, exhibits the power of an algorithmic approach. I've written the mainline algorithm (MAINLINE SECTION) in a rather general way. The number of rounds in a game is left open; the number of dice rolled and the number of sides on a die is left open; the identity of the two players is not assumed in the structure of the main algorithm. All of these game parameters are nailed down in the ALGORITHM-UTILITIES SECTION. Thus, the utilities section takes the form of "pluggable" modules which I can alter easily. I can, for instance, investigate the effect on the game of making PLAYER-1 the computer and PLAYER-2 the human—I suspect the human's strategy would change under these conditions. I could give the algorithm to a student in an introductory course in probability and ask him to supply the data array for the INITIALIZE-PLAYER-2-STRATEGY; an obvious test of his understanding would be to let someone else attempt to beat his program. All sorts of other approaches to a strategy leap to mind; to test them, we set up the program with your strategy as an automated Player 1 and my strategy as an automated Player 2, and let the program run off a few hundred games — whose strategy is best? What does "best" mean: short range success, long range success, or something else? This is the kind chain-reaction investigation that so excites me about working with algorithms instead of a specific program. So many possibilities leap right out at you from a well-designed algorithm.

In any event, that's where I have gotten to in the last few weeks. As I mentioned earlier, it feels pretty good to me, but it's difficult to be completely objective about one's own ideas; I'm looking forward eagerly to your reactions. There are a couple of aspects of my pseudo-BASIC that feel rough to me yet. One is the naming of variables— BASIC's convention is a little restrictive and, I think, tends to obscure the usage of a variable. The other area that has me really stumped right now is string handling and character manipulation. As a *programming* technique, BASIC's string variable machinery may be adequate (tho I personally have never felt comfortable with that approach) but the same machinery (at the language level, of course) for the statement of algorithms doesn't please me at all.

By the by, before I forget: you may indeed consider me a volunteer to extract algorithms from any programs you might care to send along. In this regard, I'm already working on an algorithm for LUNAR LANDER and all of the related programs. It'll be a while on that one, tho, because I want to make sure I'm on firm ground with my physical model first. Also, don't drop STAR TREK on me right away — I'm not sure I'm ready for a trauma of that magnitude just yet! Otherwise, let 'er rip and we'll see what happens.

If I can arrive at some sort of reasonable way to handle string variables in pseudo-BASIC, there are a couple of programs in my file that I'd like to write up for publication. I wrote them initially a few years ago, but they need some cleaning-up and I don't have access to BASIC now so I'd have to develop them as algorithms. One is a program that generates "plots" for those horrible Japanese Sci-Fi movies that you see on TV in the wee hours of the morning. The other program came out of an interesting course in computational linguistics I took a few years ago. The program takes Spanish words and separates them into syllables. Turns out that syllabification in Spanish is a very regular, well-defined process.

I think I've about run out of gas for now; looking forward to your reactions.

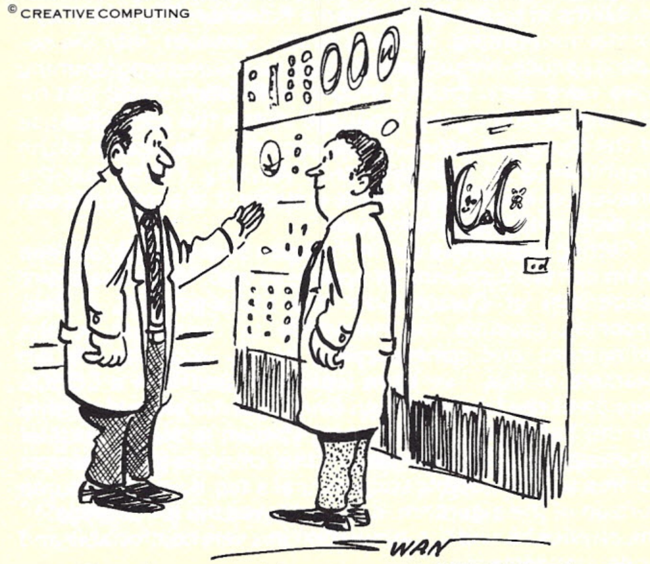
Readers: it's up to you.

Example #1 — GUESS A NUMBER

"Guess a Number" is a program in which the computer chooses a number and the user tries to guess it in as few tries as possible. The computer tells the user after each guess whether it was high or low.

Versions of the program have appeared in *Creative Computing,* Sep/Oct 1975 and Mar/Apr 1977 (see "A Musical Number Guessing Game" elsewhere in this issue), *The Best of Creative Computing — Vol. 1, PCC,* and *What To Do After You Hit Return.*

```
MAINLINE SECTION.
   START-HERE.
         REM *** PRINT INSTRUCTIONS FOR GAME.
         GOSUB PRINT-INSTRUCTIONS
GET-COMPUTER-NUMBER.
         REM *** COMPUTER "THINKS" OF A NUMBER,
         X
         LET X= INT (100 # RND (   )) + 1
         PRINT "OK, I HAVE A NUMBER. START GUESS-
         ING."
   HUMAN-GUESS.
         REM *** HUMAN MAKES A GUESS, G
         PRINT "WHAT IS YOUR GUESS?"
         INPUT G
   CHECK-GUESS.
         REM *** COMPARE GUESS TO TARGET
         IF G=X THEN GUESS-IS-CORRECT
         IF G  V  X THEN GUESS-IS-TOO-BIG
         GO TO GUESS-IS-TOO-SMALL
   GUESS-IS-TOO-SMALL.
         PRINT "TOO SMALL, TRY A LARGER NUMBER."
         GO TO HUMAN-GUESS
   GUESS-IS-TOO-BIG.
         PRINT "TOO BIG, TRY A SMALLER NUMBER."
         GO TO HUMAN-GUESS
   GUESS-IS-CORRECT.
         PRINT "YOU GUESSED IT! LET'S PLAY AGAIN."
         GO TO GET-COMPUTER-NUMBER
PROGRAM-UTILITIES SECTION.
   PRINT-INSTRUCTIONS
         PRINT "HERE ARE PLAYING INSTRUCTIONS:"
               •
               •
               •
         RETURN
```

"It makes everything easy as 3.1415926 ..."

Example #2 — NOT ONE

"Not One" was originally presented in *Creative Computing*, Nov/Dec 1974 and Mar/Apr 1975, but is now available in *The Best of Creative Computing — Vol. 1* (pp 252-253).

The game consists of 10 rounds. On each turn a player rolls a pair of dice from 1 to N times. His score is the total of all rolls, however, if any roll equals the first roll on that turn, the turn ends with a score of 0. A more complete description can be found in the above references.

```
MAINLINE SECTION.
  BEGIN-GAME.
        REM *** INITIALIZE PLAYER'S TOTAL SCORES.
        LET P1 = 0
        LET P2 = 0
  START-ROUND.
        REM *** A ROUND CONSISTS OF 1 TURN FOR
        REM *** EACH PLAYER; THERE ARE R1 ROUNDS
        REM *** IN A GAME
        FOR R = 1 to R1
        PRINT "ROUND"; R
  PLAYER-1-TURN.
        REM *** PLAYER'S FIRST ROLL ESTABLISHES
        HIS
        REM *** TARGET NUMBER; PLAYER CRAPS OUT
        REM *** IF ANY SUBSEQUENT ROLL HITS THE
        TARGET.
        GOSUB INITIALIZE-PLAYER-1-STRATEGY
  PLAYER-1-FIRST-ROLL.
        REM *** ESTABLISH PLAYER 1 TARGET
        NUMBER.
        REM *** TOTAL OF ROLL IS RETURNED IN T.
        REM *** NUMBER OF DICE ROLLED IS IM-
        MATERIAL
        REM *** AT THIS LEVEL OF THE LOGIC.
        GOSUB ROLL-AND-TOTAL-DICE
        LET T1 = T
        LET S1 = T
  PLAYER-1-SHOW-AND-TELL.
        REM *** PLAYER 1 IS TOLD THE RESULT OF
        REM *** THE ROLL AND ASKED WHAT HE
        WANTS
        REM *** TO DO NEXT.
        GOSUB TELL-PLAYER-1-THE-RESULT
        GOSUB ASK-PLAYER-1-WHAT-NEXT
        IF REPLY = "STOP" THEN PLAYER-1-TURN-ENDS
  PLAYER-1-ROLLS-AGAIN.
        REM *** PLAYER 1 HAS ELECTED TO CONTINUE
        GOSUB ROLL-AND-TOTAL-DICE
        IF T=T1 THEN PLAYER-1-CRAPS-OUT
        LET S1 = S1 + T
        GO TO PLAYER-1-SHOW-AND-TELL
  PLAYER-1-CRAPS-OUT.
        REM *** PLAYER 1 CRAPPED OUT; HIS SCORE
        REM *** FOR ROUND IS 0.
        LET S1 = 0
        GOSUB TELL-PLAYER-1-THE-RESULT
        GOSUB TELL-PLAYER-1-BAD-NEWS
  PLAYER-1-TURN-ENDS.
        REM *** PLAYER 1 HAS COMPLETED HIS TURN;
        REM *** UPDATE HIS SCORE.
        LET P1 = P1 + S1
  PLAYER-2-TURN
        REM ***  THE ALGORITHM FOR PLAYER 2's
        TURN
        REM *** IS THE SAME AS FOR PLAYER 1.
        GOSUB INITIALIZE-PLAYER-2-STRATEGY
  PLAYER-2-FIRST-ROLL.
        GOSUB ROLL-AND-TOTAL DICE
        LET T2 = T
        LET S2 = T
  PLAYER-2-SHOW-AND-TELL
        GOSUB TELL-PLAYER-2-THE-RESULT
        GOSUB ASK-PLAYER-2-WHAT-NEXT
        IF REPLY ="STOP" THEN PLAYER-2-TURN-ENDS
  PLAYER-2-ROLLS-AGAIN.
        GOSUB ROLL-AND-TOTAL-DICE
        IF T=T2 THEN PLAYER-2-CRAPS-OUT
        LET S2=S2+T
        GO TO PLAYER-2-SHOW-AND-TELL
  PLAYER-2-CRAPS-OUT.
        LET S2=0
        GOSUB TELL-PLAYER-2-THE RESULT
        GOSUB TELL-PLAYER-2-BAD-NEWS
  PLAYER-2-TURN-ENDS.
        LET P2 = P2 +S2
  END-ROUND.
        GOSUB PRINT-SCORE-FOR-ROUND
        NEXT R
  END-OF-GAME.
        GOSUB PRINT-GAME-SCORE
        STOP

ALGORITHM-UTILITIES SECTION.
  INITIALIZE-PLAYER-1-STRATEGY.
        REM *** LET PLAYER 1 INITIALIZE HIS OWN
        STRATEGY
        RETURN
        ROLL-AND-TOTAL-DICE.
        REM *** ROLL 2 DICE, PASS TOTAL THROUGH
        VARIABLE T.
        LET D1 = INT (6 * RND ( ) ) + 1
        LET D2 = INT (6 * RND ( ))+ 1
        LET T = D1 + D2
        RETURN
  TELL-PLAYER-1-THE-RESULT
        PRINT T
        RETURN
  ASK-PLAYER-1-WHAT-NEXT.
        PRINT "STOP OR ROLL AGAIN?"
        INPUT REPLY
        RETURN
  TELL-PLAYER-1-BAD-NEWS.
        PRINT "YOU GET A ZERO FOR THIS ROUND."
        RETURN
  INITIALIZE-PLAYER-2-STRATEGY.
        REM *** LOAD PROBABILITY VECTOR
        RESTORE
        FOR I = 2 TO 12
        READ Z(I)
        NEXT I
        RETURN
  TELL-PLAYER-2-THE-RESULT.
        RETURN
  ASK-PLAYER-2-WHAT-NEXT.
        LET Z (T2) = Z (T2) - 1.
        IF Z (T2) 1 THEN PLAYER-2-STOPS
        PLAYER-2-GOES-AGAIN.
        LET REPLY = "GO"
        GO TO ASK-PLAYER-2-EXIT
        PLAYER-2-STOPS.
        LET REPLY = "STOP"
        ASK-PLAYER-2-EXIT.
        RETURN
  TELL-PLAYER-2-BAD-NEWS.
        RETURN
```

Add a Kluge Harp to Your Computer

Carl Helmers

One of the most interesting computer applications is that of electronic music. This is the use of software/hardware systems to produce sequences of notes heard in a loud speaker or recorded on magnetic tape. The idea of generating music — if well done — is of necessity complex. If I want to put my favorite Mozart piano sonata into an electronic form, I'd have to record a very large number of bits in order to completely specify the piece with all the artistic effects of expression, dynamics, etc...The magnitude of the problem can be intimidating. But, never let a hard problem get in the way of fun! Simplify the music problem to one channel of melody, and you can use a virtually bare CPU with a very simple peripheral to play music.* The combination of the CPU with this simple peripheral is what I call the "Kluge Harp" — a quick and dirty electronic music kluge.

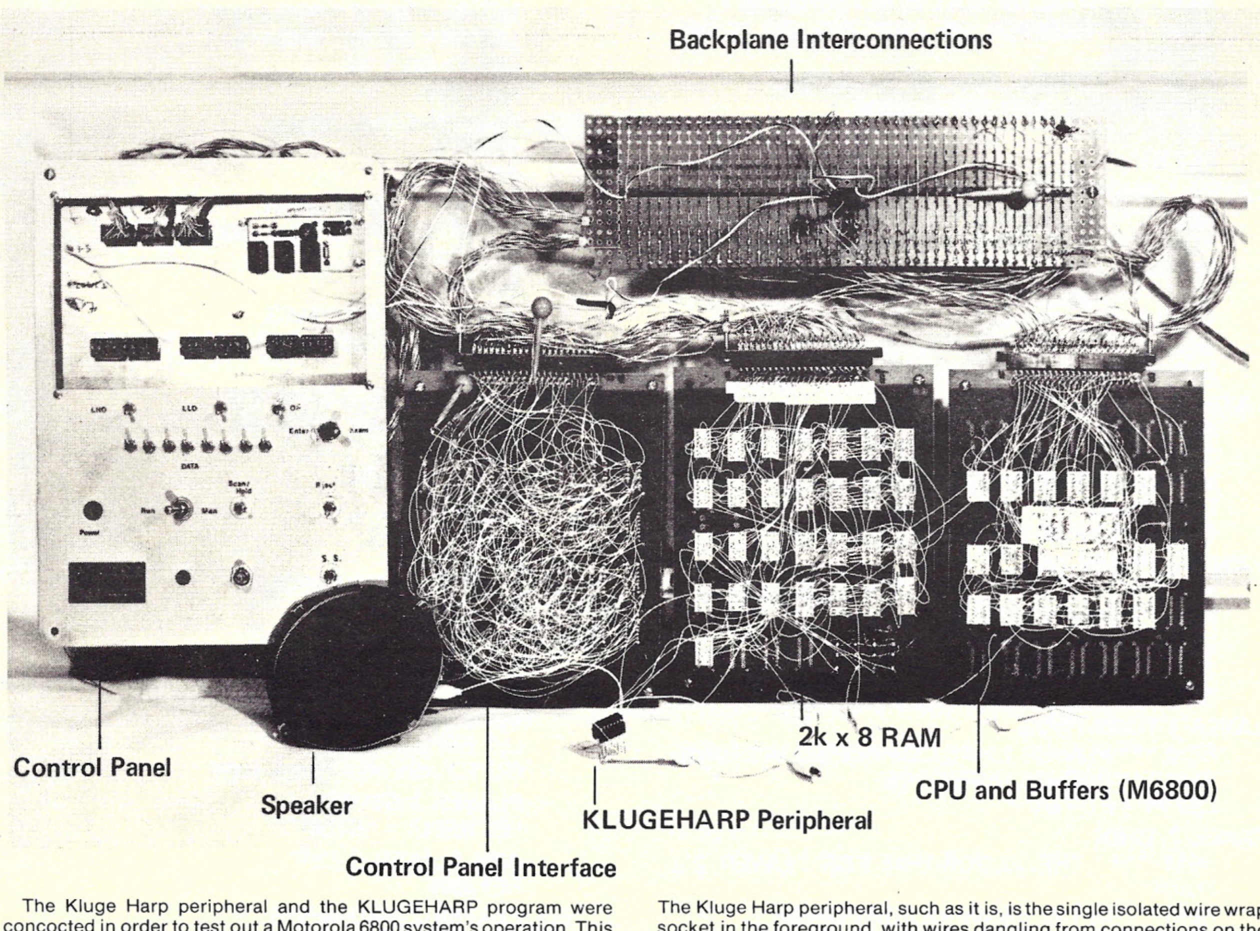
I invented this electronic music kluge to answer a specific problem: I had just gotten a new Motorola 6800 system's CPU, memory and control panel up and running. The next problem (since I wasn't using the Motorola ROM software) was to make a test program which could be loaded by hand. By combining a little imagination, my predilections for computer music systems and an evening getting the whole mess straightened out, the Kluge Harp resulted. While the program and schematic are specific to the system I was using, the *idea* can be applied to your own system just as well.

Reprinted with permission from Byte, Vol. 1, No.2. Copyright 1975 by Byte Publications, Inc., Peterborough, N.H. 03458

The Kluge Harp Hardware

The hardware of a Kluge Harp is simplicity at its essence. The peripheral is driven off two "un-used" high order address lines (I used A14 and A13), and consists of a set-reset flip flop. A program running in the computer alternately will set and reset the flip flop by referencing one or the other of two addresses. These addresses are chosen so that the address lines in question will change state, actuating the set or reset side of the flip flop. A "note" at some pitch consists of a delay loop in the program followed by instructions to change the state of

Michael Gipe



The Kluge Harp peripheral and the KLUGEHARP program were concocted in order to test out a Motorola 6800 system's operation. This photo shows a test bench mounting of the three main cards and control panel.

The Kluge Harp peripheral, such as it is, is the single isolated wire wrap socket in the foreground, with wires dangling from connections on the CPU card.

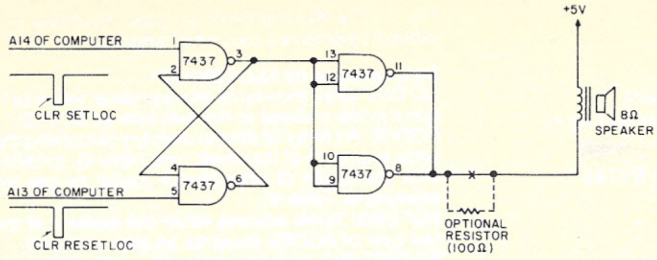


Fig. 1. The Kluge Harp Circuit ... minus computer.

the flip flop. Since the same count is used for the two halves of a complete cycle of the note, this will produce a perfect square wave. The actual music program organization is a bit more complex and is described in detail below.

Fig. 1 illustrates the hardware as implemented in my system. The 7437 circuit is used to form the NAND gate flip flop. This flip flop in turn drives a parallel combination of the two remaining 7437 gates, acting as a buffer. The output of this buffer is used to drive the speaker; an 8 Ohm 5" speaker produced more than adequate volume. (A 100 Ohm resistor in series will limit the volume level to spare the ear drums.)

Generating Music With Program Loops

Fig. 2 illustrates the basic concept of the one-channel music generator, expressed in a procedure-oriented language for compactness. The main program loop begins at line 2 of the listing — "DO FOREVER" means repeat over and over again all the lines of code down through the "END" at the same margin, found at line 17. This is the main loop used to cycle through the SCORE stored at some point in memory as pairs of note selection/length data bytes.

Lines 3 to 4 compute the "next" pointer to the SCORE — incrementing NOTER by 2. Then LNGTH is set equal to the second byte of the current pair, SCORE (NOTER+1). The length codes are taken from Table I along with note codes when you set up a SCORE, and represent a fixed interval of time for the note in question, measured as the number of cycles.

Line 6 begins a note length loop which extends to line 14. This "note length" loop repeats the generation of the note a number of times indicated by the length code just retrieved. The note generation is accomplished by delaying a number of time units (CPU states) set by the pitch code found at SCORE (NOTER), then changing the state of the output flip flop and repeating the process. The loop at lines 8-10 counts down the pitch code and has a fixed delay multiplied by the pitch code to give the time for one half cycle of the desired frequency. Lines 11 to 15 change the state of the Kluge Harp output device (0 to 1, 1 to 0) — remembering in the software location IT what the previous state was.

Generating Codes

Table 1 is a reference table of 21 notes "roughly" spaced at equal intervals on the well tempered scale. The integer numbers in the "divide ratio" column were determined using the prime number 137 as an arbitrary starting point and calculating the integer closest to the result of the following formula:

$$r_n = e^{(\ln(137)+n\ln(2)/12)}$$

Where e is the usual mathematical number 2.717 ... and the natural logarithm of x (base e) is indicated by ln(x). This is the standard mathematical calculation of the musical "well tempered" scale — the 8-bit approximation used by the Kluge Harp is not perfect by any means, but comes close enough for the purposes of this project.

n	divide ratio	hex note code	Note Length Codes (second byte of pair) 1	2	4	6	8	16	32
-10	77	4D	19	32	64	96	C8	–	–
-9	81	51	18	30	60	90	C0	–	–
-8	86	56	17	2D	5A	87	B4	–	–
-7	91	5B	16	2B	56	81	AC	–	–
-6	97	61	14	29	51	7A	A2	F3	–
-5	102	66	13	27	4D	74	9A	E7	–
-4	108	6C	12	25	49	6E	92	DB	–
-3	115	73	11	23	43	68	8A	CF	–
-2	122	7A	10	21	41	62	82	C3	–
-1	129	81	10	1F	3E	5D	7C	BA	F8
0	137	89	0F	1D	3A	57	74	AE	E8
1	145	91	0E	1C	37	53	6E	A5	DC
2	154	9A	0D	1A	34	4E	68	9C	D0
3	163	A3	0C	19	31	4A	62	93	C4
4	173	AD	0C	18	2F	47	5E	8D	BC
5	183	B7	0B	16	2C	42	58	84	B0
6	194	C2	0B	15	2A	3F	54	7E	A8
7	205	CD	0A	14	28	3C	50	78	A0
8	217	D9	09	13	25	38	4A	6F	94
9	230	E6	09	12	23	35	46	69	8C
10	244	F4	08	11	21	32	42	63	84

Table I. Kluge Harp Synthesizer pitch/length specification codes (HEX).

6800 Address	Value	
FC00	9A34	Note 1
FC02	9A34	
FC04	9A34	
FC06	9A34	
FC08	9A34	
FC0A	9A34	
FC0C	9A34	
FC0E	9A34	
FC10	7A41	Note 2
FC12	7A41	
FC14	7A41	
FC16	7A41	
FC18	664D	Note 3
FC1A	664D	
FC1C	664D	
FC1E	664D	
FC20	A331	Note 4
FC22	A331	
FC24	A331	
FC26	A331	
FC28	A331	
FC2A	A331	
FC2C	9A34	Note 5
FC2E	893A	Note 6
FC30	9A34	Note 7
FC32	9A34	
FC34	9A34	
FC36	9A34	
FC38	9A34	
FC3A	9A34	
FC3C	9A34	
FC3E	9A34	

6800 Address	Value	
FC40	5B56	Note 8
FC42	5B56	
FC44	5B56	
FC46	5B56	
FC48	5B56	
FC4A	5B56	
FC4C	5B56	
FC4E	5B56	
FC50	664D	Note 9
FC52	664D	
FC54	664D	
FC56	664D	
FC58	4D64	Note 10
FC5A	4D64	
FC5C	4D64	
FC5E	4D64	
FC60	664D	Note 11
FC62	664D	
FC64	664D	
FC66	664D	
FC68	7343	Note 12
FC6A	664D	Note 13
FC6C	7343	Note 14
FC6E	7A41	Note 15
FC70	7343	Note 16
FC72	7A41	Note 17
FC74	7A41	
FC76	7A41	
FC78	7A41	
FC7A	7A41	
FC7C	7A41	
FC7E	7A41	
FC80	(end pointer points here)	

Table II. WOLFGANG: Set the content of SCORE in memory to the codes in this table — given for the addresses of the M6800 program version — and KLUGEHARP will play four bars from the classical period.

NOTE: This program is simpleminded and not at all optimized. As a challenge to readers, figure out a way to make the notation more compact yet preserving the total length of each note.

```
1   KLUGEHARP:  PROGRAM;
2      DO FOREVER;
3         NOTER = NOTER + 2;
4         IF NOTER = NOTEND THEN NOTER = NOTESTART;
5         LNGTH = SCORE(NOTER+1); /* SECOND OF TWO BYTES */
6         DO FOR I = LNGTH TO 1 BY -1;
7            PITCH = SCORE(NOTER); /* FIRST OF TWO BYTES */
8            DO FOR J = PITCH TO 1 BY -1;
9               /* COUNT DOWN THE PITCH DELAY */
10           END;
11           IT = IT + (-127); /* SWITCH SIGN BIT OF IT */
12           IF IT  0 THEN
13              SETLOC = 0; /* SET FLIP FLOP WITH MEMORY REF */
14           ELSE
15              RESETLOC = 0; /* RESET FLIP FLOP WITH REF */
16        END;
17     END;
18  CLOSE KLUGEHARP;
```

Fig. 2. The KLUGEHARP program specified in a procedure-oriented computer language.

Data assumed by KLUGEHARP:
NOTER: 16-bit (two-byte) address value. Initialize to point to the address of the first byte of SCORE.
SCORE: An array of data in memory containing the code sequence of the music (see Table II). Initialize with the music of your heart's desire or use the example of Table II.
NOTEND: 16-bit address value, the address of the last byte of SCORE (must be an even number).
NOTESTART: 16-bit address value, the address of the first byte of SCORE (must be an even number).
SETLOC: An unimplemented address location which if referenced turns off one bit among the high order address lines, bit 14 in the author's case.
RESETLOC: An unimplemented address location which if referenced turns off one bit among the high order address lines, bit 13 in the author's case.
Data used but not initialized:
LNGTH
PITCH
IT
I,J

Address	**Data**	**Label**	**Opcode**	**Operand**	
F800	FE	KLUGEHARP:	LDX	NOTER	
F801	FA	3:			
F802	00				Add 2 to location in score
F803	08		INX		by incrementing and then
F804	08		INX		saving 16-bit new address
F805	FF		STX	NOTER	
F806	FA				
F807	00				
F808	8C	4:	CPX	#NOTEND	compare against immediate
F809	FC	NOTEND:	(last address of		
F80A	80		SCORE plus 2)		
F80B	26		BNE		Skip if not at end . . .
F80C	03		*+3+2		
F80D	CE		LDX	#NOTESTART	
F80E	FC	NOTESTART:	(first address of		otherwise recycle
F80F	00		score . . .)		
F810	FF		STX	NOTER	save in either case . . .
F811	FA				
F812	00				
F813	FE		LDX	NOTER	This is superfluous!
F814	FA				
F815	00				
F816	E6	5:	LDAB	1,X	
F817	01				
F818	5A	LENGTH:	DECB		
F819	26	6:	BNE		Skip if length remains . . .
F81A	03		*+2+3		
F81B	7E		JMP	KLUGEHARP	Restart piece
F81C	F8				
F81D	00				
F81E	A6	7:	LDAA	0,X	
F81F	00				
F820	4A	FLOOP:	DECA		
F821	26	8:	BNE	FLOOP	
F822	FD		*+2-3		
F823	86	11:	LDAA	#80	
F824	80		(-127)		
F825	BB		ADDA	IT	
F826	FA				
F827	02				
F828	2B	12:	BMI		
F829	05		*+2+5		
F82A	7F	13:	CLR	SETLOC	
F82B	B0		(address with bit 14 off . . .)		
F82C	00				
F82D	20		BRA		
F82E	03		*+2+3		
F82F	7F	15:	CLR	RESETLOC	
F830	D0		(address with bit 13 off . . .)		
F831	00				
F832	B7		STAA	IT	
F833	FA				
F834	02				
F835	7E	16:	JMP	LENGTH	
F836	F8				
F837	18				

Fig. 3. Motorola 6800 Code for KLUGEHARP program.

Data allocations for KLUGEHARP:
FA00-FA01 = Current pointer to SCORE, NOTER, which should be initialized to FC00 before starting the program.
FA02 = before starting the program.
FA02 = IT — an arbitrary initialization will do.
FA03-FFF7 = memory area available for SCORE — the example uses FC00 to FC7F and puts the relevant initializations into locations F809-F80A (NOTEND) and F80E-F80F (NOTESTART).

Note: In the label column, the numbers followed by colons (e.g., "6:") are used to indicate corresponding places in the high level language version of the program of Fig. 2.

In the system for which this program was written, all active memory is found at addresses F800 to FFFF. Thus for all normal program activity, bits A14 and A13 at the back plane of the system are logical "1". When the location SETLOC (B000) is cleared, the high order address portion changes and bit A14 goes to negative for a short time, setting the Kluge Harp flip flop. When the location A13 is cleared (D000) on an alternate cycle, address bit A13 goes to logical 0 for a short timer resetting the Kluge Harp flip flop ...

The length count columns are determined based upon the assembly language generated code for this routine, so that for each pitch, the corresponding length count column will measure a nearly identical interval of time. The formula is:

$$Lc_n = \text{time} / (\text{oh} + \text{dt}\#\text{pc}_n) \text{ where:}$$

Lc_n=n^{th} length count.
time is the total number of states for one "beat" of the music (e.g., the shortest note).
oh is the overhead of the length counting loop.
dt is the number of states the pitch count innermost loop.
pc_n is the pitch count for the nth frequency.

Table I shows the divide ratio in decimal, a hexadecimal equivalent note pitch code, and seven columns of hexadecimal length codes weighted to 1, 2, 4, 6, 8, 16 and 32 unit intervals of time. A note is placed in the score by picking a note code, putting it in an even numbered byte, then placing a length code from the same line of the table in the odd numbered byte which follows it. The actual pitches you'll get from these codes depend upon the details of the algorithm in your own particular computer and the clock rate of the computer. For the 6800 system on which Kluge Harp was first implemented, the lowest note (code F4) is approximately 170 Hz with a 500 kHz clock — and the unit interval of time is approximately 2000 CPU states or about 4 milliseconds.

The hand assembled M6800 code for the KLUGEHARP program is listed in Fig. 3. The mnemonics and notations have been taken from the *Motorola M6800 Microprocessor Programming Manual* available from the manufacturer.

While not the greatest musical instrument in the world, the Kluge Harp represents an interesting and challenging diversion. The program presented here is by no means the ultimate in music systems — and can serve as a basis for further experimentation and elaboration. Some challenges for readers: modify the program to change the frequency of the notes without changing the SCORE data; write another (longer) music program which only specifies the pitch code/length information once — and represents the score as a series of one-byte indices into the table of pitch code/length information. ■

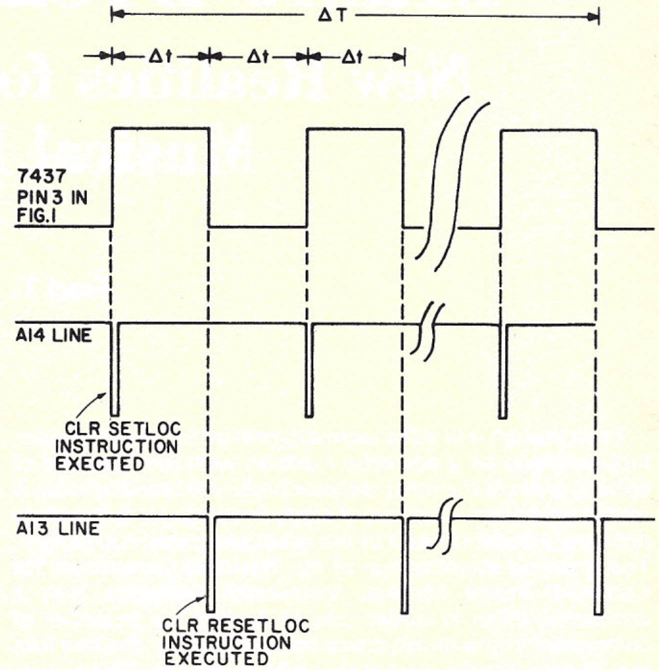


Fig. 4. Timing of the Kluge Harp Output Waveform. Δ t is the amount of time spent in the inner loop, and is set by choice of pitch codes. ΔT is the length of the note, measured as a count of half cycles at its frequency. See Table I for a consistent set of length codes.

Music Dream Machines:

New Realities for Computer-Based Musical Instruction

by
Fred T. Hofstetter*

Even though it is still a technological infant, the computer has emerged as a powerful medium with the potential of solving some of the most complex problems faced by today's music educators. Courses of study in computer technology are now a common part of the graduate music curriculum. The growing membership of the National Consortium for Computer-Based Musical Instruction indicates that a concerted effort is under way to realize the potential of computers in the music classroom. Moreover, the fact that the Norlin Music Company, which makes Moog synthesizers, Lowery organs, and Gibson guitars, is manufacturing a stand-alone music education system using minicomputer technology indicates that a major trend in music education may be under way.

At least twenty university music departments have initiated major developmental programs to produce computer-based musical materials (Jones, 1975). Preliminary reactions to these materials have been overwhelming. Teachers like them; students like them; and researchers like them.

The classroom music teacher faces many logistical problems. A well-planned presentation usually involves the use of several media, such as records, tapes, slides, transparencies, and duplicated handouts. The physical preparation of these materials, as well as the actual manipulation of them during class, can be cumbersome and time-consuming for both teacher and student. The teacher can encounter difficulties in procuring desired materials for the presentation, and unless time is taken to locate the material, the students have to settle for a second-rate presentation. Students remember many occasions when valuable class time was lost while the teacher tried to find the right place on a record or a tape in order to play a musical example.

If a music teacher were asked to state the requirements of a classroom presentation "dream-machine," the response would be a device capable of displaying musical notation, showing slides, playing recordings, and maybe even generating some new examples for aural training. Such a machine was dreamed about in the 1960's by Professor Donald Bitzer (1961) at the University of Illinois. He made it a reality; it is called PLATO, and it is now a product of the Contral Data Corporation (1976). PLATO is a comprehensive computer-based education system which has been used in over seventy-one subject areas, and a variety of special presentation devices are available in different subjects.

*The University of Delaware, Newark, DE

For all subjects there is a basic PLATO display unit which contains a screen upon which graphics (like musical notation) can be drawn, a random-access microfiche projector which can show slides on the screen, a typewriter keyboard through which one can communicate with the computer, and a touch panel which allows students to answer questions by touching pictures or words on the screen. Two special devices are used in music programs. One is a random-access audio device which can play any segment of a pre-recorded magnetic audio disk, and the other is a four-voice synthesizer which can be played by the computer. All of these components combine to produce a single machine which can display musical notation, show slides, play recordings, and generate new aural examples.

Figure 1. PLATO terminal with microfiche slide projected on the display screen. On top of the terminal is a random access audio unit, and the small box on top of that is the 4-voice synthesizer.

One of the greatest benefits of computer-based education is the return of personal warmth to the classroom made possible by releasing the teacher of mechanistic duties.

A super-medium like PLATO can put an end to the problems of finding the right materials because vast numbers of recorded examples, slides, and visual displays can be stored and randomly accessed at a split-second's notice in the computer. Because it has the ability to present visual displays, play aural examples, ask students questions, record student responses, and interact with students on the basis of their responses, the computer is an ideal medium for drill and practice. Due to the limited amount of time the teacher spends with students in the classroom, it is difficult to maintain a good balance between concept development and drill-and-practice activities. In trying to meet course objectives by their intended deadlines, teachers realize that their students often do not get enough practice. Teachers also know that when they devote time to classroom drills, the individual differences among students create an environment in which some students are bored because they have already mastered the material, and others are hopelessly lost because the exercises are too difficult for them. It is now possible to take drill-and-practice out of the classroom and have it done in the computer laboratory where each student receives an individualized course of study based on the student's personal learning needs. And by means of the data keeping facilities of the computer, the teacher can get periodic reports on the progress and special problems of each student.

When drill-and-practice is done in a self-paced computer-based environment, learning is more efficient. Class time formerly devoted to drills can be spent in concept development and in more creative aspects of music. The role of the teacher becomes more human. Mitzel (1972) has noted that one of the greatest benefits of computer-based education is the return of personal warmth to the classroom made possible by releasing the teacher of mechanistic duties.

The greatest proponents of computer-based musical instruction are the students themselves. It has been demonstrated that with the aid of the computer they can increase their musical sensitivities, make higher grades, spend less time completing course requirements, and rediscover the intrinsic joy of learning. Illustrations of these attributes can be seen in several schools throughout the country. At Penn State University, Diehl (1971, 1973) demonstrated that students can develop more sensitivity and accuracy in the recognition and performance of articulation, phrasing, and rhythm. An experiment was set up in which 25 junior high school students who had taken private woodwind or brass study for at least three years were tested on their sensitivity and ability to perform articulation, phrasing, and rhythm. After this pre-test they participated in a five-week computer program in which an IBM 1500 computer was used to display musical notation, play pre-recorded musical examples, and ask questions about the articulation, phrasing, and rhythm of the musical examples. The students used the computer one hour each week. At the end of the five-week program, the students were given the test again. Table 1 gives the mean test scores for the pre-and post-tests. It is interesting to note that the computer led to a 55% gain in the students' abilities to *perform* music.

TABLE 1

Summary of Student Scores in a Computer-Based Program for Improving Articulation, Phrasing, and Rhythm of Intermediate Instrumentalists

Group Averages	Listening Test Means	Performance Test Means
Pre-Test (N=25)	62.8%	38.8%
Post-Test (N=25)	92%	94%
Gains	30%	55.2%

At the University of Delaware, a computer-based ear-training system has led to significant improvement of instruction in core music theory courses. Ear-training students are scoring a full letter-grade higher than they did before. The system is called GUIDO (Hofstetter, 1975), named after the eleventh century monk and music educator who invented the staff and the solfeggio syllables *do, re, mi, fa, sol,* and *la.* GUIDO is a acronym for *G*raded *U*nits for *I*nteractive *D*ictation *O*perations. These units are stored in the computer, and they contain a complete curriculum for aural drill-and-practice in intervals, melody, harmony, rhythm, and chord-qualities. GUIDO is being implemented on a variety of computers. To date, its best operating environment is on the PLATO system, because PLATO's touch-sensitive display screen makes it possible to run the program by merely touching musical symbols on the screen. It is not necessary to type on the keyset.

Figure 3 shows a sample display from the intervals program. By studying this display the basic features of the GUIDO system can be understood. At the top are two rows of boxes which contain the names of musical intervals. When the student wants to hear an interval, all he has to do is touch one of the boxes. When he does, the box lights up and the interval designated by the box is played by the computer-controlled synthesizer. Conversely, when the student is going through one of GUIDO's formal units, the computer plays an interval, and the student responds by touching the box which contains the interval he thinks was played.

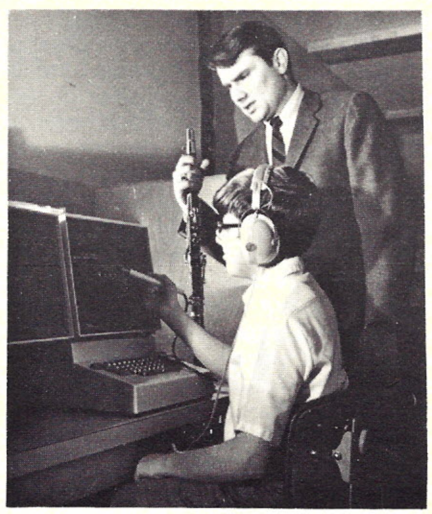


Figure 2. Professor Ned Diehl observes a clarinet student using the Penn State Instrumental Music System.

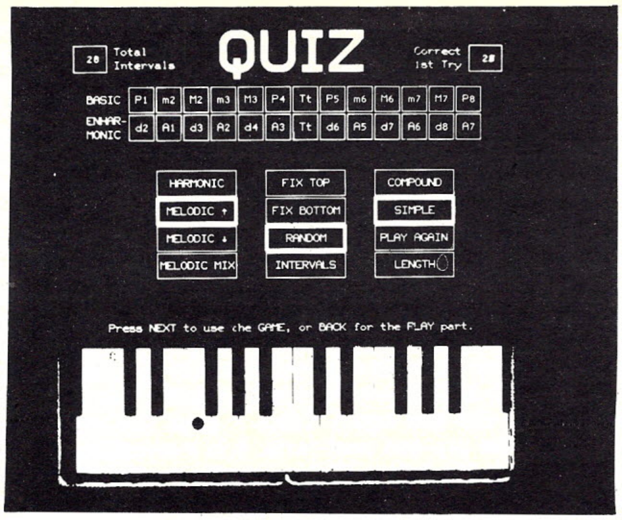


Figure 3. Touch-Sensitive Display from the GUIDO Ear-Training System.

Underneath the interval names are three columns of teacher or student control boxes. These boxes are used to control the way in which dictation is given. The teacher can preset them for the students, or the teacher can allow the students to set them at will. The first column of boxes allows for the intervals to be played as harmonic, melodic down, or melodic intervals up and down. The second column gives the option of being able to fix the top or bottom note of the intervals, or to have them selected at random. The box marked "intervals" allows the student to eliminate intervals from the boxes at the top of the screen, so that only some of the intervals will be played. In the third column of boxes the student can select compound or simple intervals, can have an interval played again, and can change the length of time the intervals last. Finally, there is a keyboard at the bottom of the screen. When intervals are played in formal units one of the notes of each interval is shown on the keyboard, and the student is asked to touch the other note played in the interval. In this way, students are quizzed on the spelling as well as on the aural recognition of intervals.

In addition to liking computers because they make learning more efficient and improve grades, students like the interactive help which the computer gives them in analyzing music. One example is the combination of tutorial and analytical programs being developed on PLATO by Wittlich (1976) at Indiana University. These programs are based on *set theory*, the analytical procedures defined by Allen Forte (1973) for describing the pitch organization of atonal music. Wittlich is writing programs to teach music fundamentals using pitch class concepts of set theory. Topics include the definition of set, normal form, numerical set, transposition, simple inversion, inversion relative to zero, transposed inversion, interval class, and interval vectors, A very time-consuming part of set theory is the computation of normal form, interval vector, and set name for each pitch set. In fact, many people do not use these procedures because of the time involved. James Trueblood, a former Indiana student and now a programmer at Delaware, has written a program which allows the student to type in the pitch set using either letters or numbers, and then, in less than .3 second, the normal form, vector, and set name are shown on the display screen.

Figure 4 contains a sample display from the set-theory program. In the middle of the display can be seen where the computer asked for a pitch set. After the student typed in C# Eb G B, the computer immediately responded with the rest of the display which shows that the normal form is 0 2 4 8, the interval vector is 0 2 0 3 0 1, and the pitch-class name is 4-24 (12).

The computer-based music education system can present visual and aural stimuli, record responses, and perform data analysis in one automatic process.

Students also like the control which the computer gives them in writing compositions for electronic music synthesizers. Instead of struggling with dials, switches, and patch cords, the students can write computer programs which automatically control the synthesizer and create desired sounds with much less effort. In this way students can concentrate more on the organization and meaning of their music. An example of this application is the Iowa State Computerized Music System (ISMUS), developed under the direction of Gary White. This system allows students in music, computer science, and engineering to interact in the creation of musical compositions played on an ARP synthesizer under the control of a PDP-8 minicomputer.

The last and most obvious reason why students like computer-based education is because it is just plain fun. Even the circle of fifths can be fun. Figure 6 shows the display screen of a game developed under the direction of Professor David Peters (1976), head of the PLATO muisc project at the University of Illinois. Named "Keyspinner," this game calls for two players. The display consists of the circle of fifths with a needle or "spinner" inside it. The players take turns spinning the needle, and when it stops on a key signature (e.g. 5 sharps), the player has to identify the key of that signature (in this case B) as soon as possible. The computer keeps track of how long it takes for each player to answer correctly, and the player with the best score wins.

Musicians have also found the computer to be a powerful tool for pedagogical and experimental research. In the past attempts to measure the effectiveness of instructional strategies and the nature of musical learning have been hampered by the need to use manual techniques for saving and recording data. The amount of time which this requires per subject has resulted in the use of small populations when large populations would have been desired, and it has limited the number of experiments which music educators have been able to do. Of course, computers have been around for quite some time, and musicians have been using them to analyze data. What is different is that whereas in the past experimenters had to present stimuli and manually record and encode responses for data analysis, now the

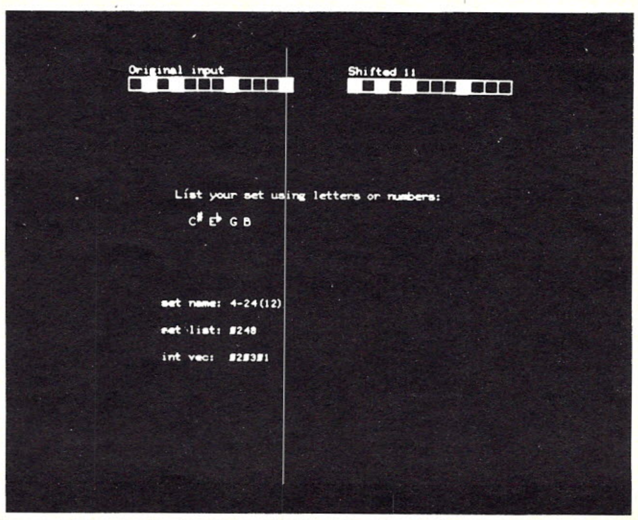


Figure 4. Sample display from Jim Trueblood's set theory program

Figure 5. Professor Gary White (seated) composing on the Iowa State Computerized Music System

computer-based music education system can present visual and aural stimuli, record responses, and perform data analysis in one automatic process. This means that the amount of experimentation can be greatly increased. It means that music educators can spend more time asking questions, and less time shuffling papers.

Another advantage is the exact control which the computer has over the sound source. If the researcher wants to play a sawtooth wave at 50 db for .3 second, he can accomplish this by giving the computer a few simple instructions. This immediate, precise control of the sound source reduces the time needed to set up an experiment, and it guarantees that during the course of an experiment there will be no unintentional variation of the sound source which could bias the results.

To date, the best example of the use of a computer-based music education system for experimental research is the study of harmonic and melodic interval recognition conducted at Stanford by Killam, Lorton, and Schubert (1976). Using a PDP-10 computer, a model 33 KSR teletype, and a computer-controlled Thomas model 145 solid-state organ, a system was designed to play intervals, ask subjects for the names of intervals played, and record their responses. Six sets of intervals were played for each of fifteen subjects. One set consisted of 48 harmonic intervals played for .2 second. The second set consisted of 48 ascending melodic intervals played for .2 second. The third set consisted of 48 descending melodic intervals played for .2 second. The fourth, fifth, and sixth sets were the same as sets one, two, and three except that they were played for only .1 second. Within each set, intervals were selected at random, and they ranged in size from a minor second to a perfect octave.

The results of this study showed that some beliefs on which teaching methods are based may be myths. First of all, when students encounter difficulty in hearing intervals in classroom dictation drills, one of the first techniques the teacher will try is slowing down the speed of dictation. However, in this study there was no significant difference in student performance when intervals were played for .1 second, when the average correct recognition was 76%, or for .2 second, when the correct recognition was 77%. Second, it is commonly believed that the perfect octave is extremely easy to recognize, that it is such a "sure thing" that little class time needs to be spent studying it. In the Stanford experiment, however, it was shown that the perfect octave is not such a sure thing. In fact, it was missed an average of 12% of the time. Moreover, seven subjects found the octave more difficult to recognize than some other interval. Finally, the study showed that there are differences of recognition related to the mode of presentation used. Table 2 contains a confusion matrix summary for modes of presentation. In this table it can be seen that the most frequently used wrong answers vary as a function of mode of presentation. For example, whereas the subjects confused the minor sixth (m6) with the perfect fourth (p4) in simultaneous and ascending intervals, they confused it more often with the major sixth in descending intervals.

Information about student learning patterns is needed by writers of textbooks so that their materials can be presented in the best order, by teachers so that they can deal more effectively with their students, and by the students themselves so that they can be aware of and try to avoid common pitfalls of musical perception. With the emergence of the computer-based music education system a comprehensive body of knowledge about musical learning will be obtained whereby these needs can be fulfilled.

It has been shown how computer-based techniques are improving music education for the teacher, the student, and the researcher. They have generated such wide-spread interest that vendors are beginning to market programs for music instruction. It was mentioned that the Norlin Music Company is manufacturing the first stand-alone music education system. It is doubly encouraging to see this support from vendors at a time when the cost of computer hardware is rapidly decreasing. One does not need to go out

TABLE 2

Confusion Matrix Summary for Modes of Presentation in the Stanford Study of Interval Recognition

Most Frequently Used Wrong Answers

Stimulus	Simultaneous	Ascending	Descending
m2	M2	M2	M2
M2	m2	m3	m2
m3	M3	M3	M3
M3	m3	m3	m3
P4	M3 & P5	P5	P5
T	m3 & M7	P4	P4
P5	P4	P4	P4
m6	P4	P4	M6
M6	m6	m7	m6
m7	M7	M7 & M6	M7
M7	T	m7	m7
P8	M7	M7	P5

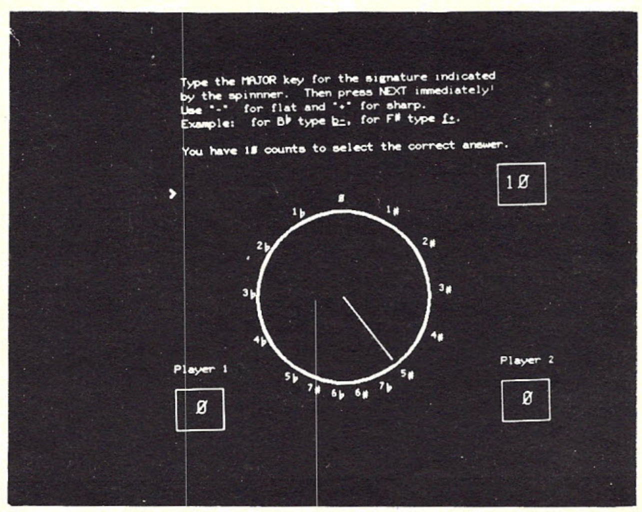


Figure 6. Display of the Keyspinner game developed by Professor David Peters at the University of Illinois.

on a limb to predict that broad-based implementation of computer-based musical instruction will occur before the end of the next decade. ■

Bibliography

Bitzer, D. L., P. Braunfeld, and W. Lichtenberger. PLATO: An automatic teaching device. *IRE Transactions on Education,* E-4 (December, 1961), 157-161.

Control Data Corporation. *Control DataPLATO system overview.* St. Paul: Control Data Corporation Publications, 1976

Diehl, Ned C. Computer-assisted instruction and instrumental music: implications for teaching and research. *Journal of Research in Music Education,* 1971, *19,* 299-306.

Diehl, Ned C., and Ray H. Ziegler. Evaluation of a CAI program in articulation, phrasing, and rhythm for intermediate instrumentalists. *Council for Research in Music Education,* 1973, *31,* 1-11.

Forte, Allen. A program for the analytic reading of scores. *Journal of Music Theory,* 1966, *10*(2), 330-64.

Hofstetter, Fred T. GUIDO: An interactive computer-based system for improvement of instruction and research in ear-training. *Journal of Computer Based Instruction,* 1975, *1,* 100-106.

Jones, Morgan. *Computer-assisted instruction in music: a survey with attendant recommendations.* Ph.D. dissertation, University of Iowa, 1975.

Mitzel, Harold E. "The Potential Contribution of Computers to Instruction Reform." *Alternative Futures in American Education,* Appendix 3 to Hearings on H. R. 3606 and Related Bills to Create a National Institute of Education Before the Select Subcommittee on Education, January, 1972.

Peters, G. David. The fourth revolution in education and music: the PLATO music project. Paper presented at the Music Educators' National Conference, Atlantic City, March 12, 1976.

Wittlich, Gary. Computers and music instruction at Indiana University (Bloomington). *Proceedings of the First Annual Meeting of the NCCBMI* (University of Delaware: Music Department, 1976).

"You say you're an expert on computer technology?"

Computer Music Bibliography

John Snell

Digital circuits may be used for controlling analog synthesizers, direct digital synthesis, composing music, analyzing (or tracking several parameters of) traditional musical instruments and the voice, spacial movement of sounds, and processing of musical sounds (filtering, reverberation, choral effects, etc.). I hope the following list of articles and books will help some of you to develop systems which are capable of making music enjoyable even by master musicians. This list is relatively short, and includes only a "taste" of relevant topics not specifically about digital music. For a more comprehensive, well-organized listing see the bibliography from *Electronotes* (a fine electronic music periodical edited by Bernie Hutchins).

J. Allen, "Computer Architecture for Signal Processing," *IEEE,* Vol. 63, No. 4; April, 1975.

W. Apel, *Havard Dictionary of Music,* Halliday Lithograph, 1969.

J. Backus, *The Acoustical Foundations of Music,* W.W. Norton, 1969.

J.K. Baker, "The DRAGON System—An Overview," *IEEE,* Vol. ASSP-23, No. 1; Feb., 1975.

S.C. Bass, and B.J. Leon, "Designers' Guide to Digital Filters," *EDN,* Jan. - June, 1974 (6 issues).

J.W. Beauchamp & H. Von Foerster, editors, *Music by Computer,* Wiley, 1969.

J.W. Beauchamp, "Analysis and Synthesis of Cornet Tones Using Nonlinear Interharmonic Relationships," *JAES,* Vol. 23, No. 10; 1975.

J.W. Beauchamp, "A Computer System for Time-Vatiant Harmonic Analysis and Synthesis of Musical Tones," *Music by Computers,* 1969.

J.W. Beauchamp, "Additive Synthesis of Harmonic Musical Tones," *JAES,* Vol. 14, No. 4, p. 332; 1966.

L. Beranek, "Digital Synthesis of Speech and Music," *IEEE Transactions on Audio,* Vol. AU-18, No. 4; Dec., 1970.

S. Bertram, "Frequency Analysis Using the Discrete Fourier Transform," *IEEE Transactions on Audio and Electroacoustics,* Vol. AU-18, No. 4; Dec., 1970.

B.A. Blesser, K. Baeder, & R. Zaorski, "A Real-Time Digital Computer for Simulating Audio Systems," *JAES,* Vol. 23, No. 9, p. 698; 1975.

B.A. Blesser, "An Investigation of Quantization Noise," *JAES,* Vol. 22, No. 1, p. 20; 1974.

B.A. Blesser, & F. Lee, "An Audio Delay System Using Digital Technology," *JAES,* Vol. 19, No. 5, p. 393; May, 1971.

E. Blom, editor, *Grove's Dictionary of Music and Musicians,* St. Martin's Press.

D. Bohn, editor, *Audio Handbook,* National Semiconductor Corp., Santa Clara, Cal.; 1976.

Research Engineer, People's Computer Co. Box 310, Menlo Park CA 94025; 415 323-3111

R.W. Burhams, "Digital Tone Synthesis," *JAES,* Vol. 19, No. 8; Sept., 1971.

R.W. Burhams, "Single-Bus Keyboard Control for Digital Musical Instruments," *JAES,* Vol. 19, No. 10, p. 865; Nov., 1971.

R.W. Burhams, "Simplified Educational Music Synthesizer," *JAES,* Vol. 19, No. 2, p. 127; Feb., 1971.

R.P. Ceely, "A Composer's View of MITSYN," *AES Preprint No. 811 (M-1),* Oct., 1971.

H. Chamberlin, "Fourier Series Waveform Generator, Part 1," *EN,* Vol. 5, No. 39, p. 2; May, 1974.

J.M. Chowning, J.M. Grey, L. Rush, & J.A. Moorer, *Computer Simulation of Music Instrument Tones in Reverberant Environments,* Stanford University Dept. of Music Report No. STAN-M-1; June, 1974.

J.M. Chowning, "The Simulation of Moving Sound Sources," *JAES,* Vol. 19, No. 1, p. 2; Jan., 1971.

J.M. Chowning, "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation," *JAES,* Vol. 21, No. 7, p. 526; Sept., 1973.

J.M. Chowning, "Stanford Computer Music Project," *NW,* No. 1; 1972.

W. Cochran, J. Cooley, D. Favin, H. Helms, R. Kaenel, W. Lang, G. Maling Jr., D. Nelson, C. Rader, P. Welch, "What is the Fast Fourier Transform," *IEEE,* Vol. 55, No. 10, p. 1664; Oct., 1967

R. Crochiere, & P. Penfield, Jr., "On the Efficient Design of Bandpass Digital Filter Structures," *IEEE,* Vol. ASSP-23; 1975.

M.G. Crosby, "Carrier and Side-Frequency Relations with Multi-Tone Frequency or Phase Modulation," *RCA Review,* Vol. 3; 1938.

D.R. Curtis, "A Monolithic Voltage-Controlled Amplifier Employing Log-Antilog Techniques," *JAES,* Vol. 24, No. 2; March, 1976.

D. De Kold, "Binary Division Produces Harmonic Frequencies," *Electronics,* Dec., 1972.

J.L. Divilbliss, "The Real-Time Generation of Music with a Digital Computer," *JMT,* No. 8, p. 99; 1964.

J. Dubnowski, R. Schafer, & L.R. Rabiner, "Real-Time Digital Hardware Pitch Detector, *IEEE,* Vol. ASSP-24, No. 1; Feb., 1976.

J.L. Flanagan & L.R. Rabiner, editors, *Speech Synthesis,* Dowden, Hutchinson, & Ross; 1973.

A. Forte, *Tonal Harmony in Concept and Practice,* Holt, Rinehart, & Winston; 1962.

N.V. Franssen & L. Van der Peet, "Digital Tone Generation for a Transposing Keyboard Instrument," *Philips Tech. Rev.,* No. 31, p. 354; 1970.

M.D. Freedman, "Technique for Analysis of Musical Instrument Tones," Ph.D. dissertation, University of Illinois, Urbana; 1965.

M.D. Freedman, "Analysis of Musical Instrument Tones," *JASA,* No. 41, p. 793; 1967.

M.D. Freedman, "A Method for Analyzing Musical Tones," *JAES,* Vol. 16, No. 4, p. 419; Oct., 1968.

S. Freeny, "Special-Purpose Hardware for Digital filtering," *IEEE,* Vol. 63, No. 4; April, 1975.

R. Gabel, "A Parallel Arithmetic Hardware Structure for Recursive Digital Filtering," *IEEE,* Vol. ASSP-22; Aug., 1974.

J. Gabura & G. Ciamaga, "Computer Control of Sound Apparatus for Electronic Music," *AES preprint No. 520;* 1967.

S. Gill, "A Technique for the Composition of Music in a Computer," *Computer Journal,* Vol. 6, No. 2, p. 129; July, 1963.

J.M. Grey, *An Exploration of Musical Timbre,* Stanford University Dept of Music Report No. STAN-M-2; Feb., 1975.

P. Grogono, "MUSYS: Software for an Electronic Music Studio," *Software—Practice and Experience,* No. 3, p. 369; 1973.

H.L.F. Helmholtz, *On the Sensations of Tone as a Physiological Basis for the Theory of Music,* translation—originally written in German in 1863—and additions by A.J. Ellis, Dover, N.Y.; 1954.

W. Henke, *Multiple Interactive Tone Synthesis System,* Research Laboratory of Electronics, M.I.T.; Oct., 1971.

L. Hiller & L. Isaacson, "Musical Composition with a High-Speed Digital Computer," *JAES,* Vol. 6, p. 154; July, 1958.

L. Hiller, "Computer Music," *Scientific American,* p. 109; Dec., 1959.

L. Hiller, "A Review of Decca Recording DL-9103, 'Music from Mathematics'," *IEEE,* Vol. 51, No. 3, p. 538; March, 1963.

L. Hiller, "Musical Applications of Electronic Digital Computers," *Gravesano Review,* pp. 26-28; Nov., 1965.

C. Hovey & D. Seamans, "A Polyphonic Keyboard for a Voltage-Controlled Music Synthesizer," *JAES,* Vol. 23, No. 6, p. 459; 1975.

K. Huehne, "Programmable ROM's Offer a Digital Approach to Waveform Synthesis," *EDN,* Aug., 1972.

B. Hutchins, "B-004 Sourcelist of Electronic Music and Musical Engineering," *EN,* 1975.

B. Hutchins, *Musical Engineer's Handbook.* (*EN* address) 1975.

B. Hutchins, "The Frequency Modulation Spectrum of an Exponential Voltage-Controlled Oscillator," *JAES,* Vol. 23, No. 3, p. 200; 1975.

B. Hutchins, "Application of a Real-Time Hadamard Transform Network to Sound Synthesis," *JAES,* Vol. 23, No. 7, p. 558; 1975.

B. Hutchins, "Digital Harmonics 1, Waveshaping 2," *EN,* Vol. 2, No. 12, p. 5; Sept., 1972.

B. Hutchins, T. Mintner, & C. Anderton, "Digital Harmonics 2," *EN,* Vol. 2, No. 13, p. 12; Oct. 1972.

B. Hutchins, "Analog Delay for Musical Engineering," *EN,* Vol. 7, No. 56; Aug., 1975.

IEEE, Vol. 63, No. 4, special issue on digital signal processing; April, 1975.

F. Itakura, & S. Saito, "Digital Filtering Techniques for Speech Analysis and Synthesis," in Conf. Rec., 7th Int. Congr. Acoustics, Paper 25C No. 1; 1971.

E.V. Jansson, & J. Sundberg, "Long-Time-Average-Spectra Applied to Analysis of Music," *Acustica,* Vol. 34, No. 1; Nov., 1975.

L.B.W. Jolley, *Summation of Series,* Dover, N.Y.; 1961.

P. Knowlton, "Capture and Display of Keyboard Music," *Datamation,* P. 56; May, 1972.

S. Kriz, "A 16-Bit A-D-A Conversion System for High Fidelity Audio Research," IEEE Symposium on Speech Recognition, p. 278; April, 1974.

P. Lehman, "Harmonic Structure of the Tone of the Bassoon," *JASA,* Vol. 36, No. 9, p. 1649; Sept., 1964.

V. Lesser, D. Fennell, L. Erman, & R. Reddy, "Organization of Hearsay II Speech Understanding System," *IEEE,* Vol. ASSP-23, No. 1; Feb., 1975.

T. Letowski, "Difference Limen for Nonlinear Distortion in Sine Signals and Musical Sounds," *Acustica,* Vol. 34, No. 2; Dec., 1975.

H. Lincoln, editor, *The Computer and Music,* Cornell University Press; 1970.

J. Link, Theory and Tuning: *Aron's Meantone Temperament and Marpsurg's Temperatment "I",* Tuners Supply Co., Boston; 1969.

B. Lin, & A. Peled, "A New Hardware Realization of High Speed Fast Fourier Transformers," *IEEE,* Vol. ASSP-23, No. 6; Dec., 1975.

H.C. Longuet-Higgins, & M.J. Steedman, *On Interpreting Bach in Machine Intelligence VI;* B. Meltzer, & D. Michie, editors, Edinburgh University Press, Edinburgh; p. 221; 1971.

D. Luce, "Description of a Real-Time Multipartial Waveform Analyzer-Synthesizer," AES preprint No. 611; 1968.

D. Luce, "Dynamic Spectrum Changes of Orchestral Instruments," *JAES,* Vol. 23, No. 7; 1975.

D. Luce, & M. Clark, "Physical Correlates of Brass Instrument Tones," *JASA,* Vol. 42, p. 1232; 1967.

M. Mathews, *The Technology of Computer Music,* M.I.T. Press, Boston; 1969.

M. Mathews, et. al., "Computers and Future Music," *Science,* Vol. 183, No. 4122; Jan. 25, 1974.

N. Mathews, "The Digital Computer as a Musical Instrument," *Science,* Vol. 142, p. 553; 1963.

M. Mathews, & J. Kohut, "Electronic Simulation of Violin Resonances," *JASA,* No. 53, p. 1620; 1973.

C.A. McGonegal, L.R. Rabiner, & A.E. Rosenberg, "A Semi-automatic Pitch Detector," *IEEE,* Vol. ASSP-23, No. 6; Dec., 1975.

N.J. Miller, "Filtering of Singing Voice Signal from Noise by Synthesis," Ph.D. thesis; University of Utah, Dept. of Computer Science; May, 1973.

N.J. Miller, "Pitch Detection by Data Reduction," *IEEE,* Vol. ASSP-23, No. 1, pp. 72-78; Feb., 1975.

F.R. Moore, & M. Mathews, "Grove—A Computer Program for Real-Time Music and Sound Synthesis," *Communications of the ACM,* Vol. 13, No. 12; Dec., 1970.

J.A. Moorer, "The Optimum Comb Method of Pitch Period Analysis of Continuous Digitized Speech," *IEEE,* Vol. ASSP-22, No. 5, p. 330; Oct., 1974.

J.A. Moorer, "Music and Computer Composition," *Communications of the ACM,* Vol. 15, No. 2; Feb., 1972.

J.A. Moorer, *On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer,* Stanford University Dept. of Music Report No. STAN-M-3; May, 1975.

J.A. Moorer, *The Synthesis of Complex Audio Spectra by Means of Discrete Summation Formulae,* Stanford University Dept. of Music Report No. STAN-M-5; Dec., 1975.

J.A. Moorer, *On the Loudness of Complex Time-Variant Tones,* Stanford University Dept. of Music Report No. STAN-M-4; Feb., 1975.

NUMUS West, No. 4; 1973. (computer music issue)

T.H. O'Beirne, "Computer Program Which Plays Music by Microtones," *Computer Journal,* Vol. 13, No. 4, p. 350; 1970.

A.V. Oppenheim, & R.W. Schafer, *Digital Signal Processing,* Prentice Hall; 1975.

W. Piston, *Counterpoint,* W.W. Norton Publishers.

W. Piston, *Orchestration,* W.W. Norton Publishers.

G. Plitnik, & W. Strong, "Digital Filter Technique for Synthesis of Bassoon Tones," *JASA,* No. 47, p. 131; 1970.

R. Plomp, "Timbre as a Multidimensional Attribute of Complex Tones," *Frequency Analysis and Periodicity Detection in Hearing,* edited by R. Plomp & G.F. Smoorenburg; A.W. Sijthoff, Leiden; 1970.

L. Rabiner & B. Gold, *Theory and Applications of Digital Signal Processing,* Prentice-Hall, N.J.; 1975.

L. Rabiner & C. Rader, editors, *Digital Signal Processing,* IEEE Press; 1972.

J.C. Risset & M. Mathews, "Analysis of Musical Instrument Tones," *Physics Today,* Vol. 22, No. 2, p. 23; 1969.

J.C. Risset, "Computer Study of Trumpet Tones," Bell Telephone Lab, Murray Hill, N.J.; 1966.

P. Samson, "Digital Signal Synthesizer," Systems Concepts, San Francisco, Cal.

P. Samson, "Digital Synthesizer—Brief Functional Description," Systems Concepts, San Francisco, Cal.

P. Samson, "Digital Synthesizer—Programming Specifications," Systems Concepts, San Francisco, Cal.

S. Saunders, "FM The Easy Way," Carnegie Mellon University Computer Science Dept & Xerox PARC; 1974.

R.A. Schaefer, "Digital Generation of Equal Temperament," *IEEE,* Vol. AASP-23, No. 4; Aug., 1975.

M. Schroeder, "Natural Sounding Artificial Reverberation, *JAES,* Vol. 10, p. 219; 1962.

M. Schroeder, "Models of Hearing," *IEEE,* Vol. 63, No. 9; Sept., 1975.

R. Schumacher, "Self-Sustained Musical Oscillators," paper delivered at the 91st meeting of the Acoustical Society of America; April, 1976.

J. Sennott, "Efficient Digital Conversion of Audio Signals," AES preprint No. 970 (M-4); Sept., 1974.

W. Slawson, Syntal II; *A Speech Oriented Computer Synthesizer Revisited,* Dept. of Music, University of Pittsburgh, Pa.

W. Slawson, "Vowel Quality and Musical Timbre as Functions of Spectrum Envelope and Fundamental Frequency," *JASA,* No. 43, p. 87; 1968.

G.W. Small, "Synthesis of the Musical Scale Using Non-integer Frequency Division," *JAES,* Vol. 21, No. 4, p. 261; May, 1973.

G.W. Small, "Rate-Feedback Binary Counters in Musical Scale Generation," *JAES,* Vol. 21, No. 9, p. 702; Nov., 1973.

L. Smith, "Score—A Musician's Approach to Computer Music," *JAES,* Vol. 20, No. 1; Jan./Feb., 1972.

L. Smith, "Editing and Printing Music by Computer," *JMT,* p. 292; Fall, 1973.

S. Smith, "Computer Music in 1972," *Computers and Automation,* Oct., 1972.

G. Steinke, "Experimental Music with the Subharchord Subharmonic Sound Generator," *JAES,* Vol. 14, No. 2, p. 140; April, 1966.

S.S. Stevens & H. Davis, *Hearing—Its Psychology and Physiology,* Wiley, N.Y.; 1938.

T.G. Stockham, Jr., "A-D and D-A Converters: Their Effect on Digital Audio Fidelity," AES Preprint No. 834 (D-1), Oct., 1971.

W. Strong & M. Clark, "Synthesis of Wind-Instrument Tones," *JASA,* Vol. 41, p. 39; 1967.

W. Strong, & M. Clark, "Perturbations of Synthetic Orchestral Wind-Instrument Tones," *JASA,* No. 41, p. 277; 1967.

R. Talambiras, "Digital-to-Analog Converters: Some Problems in Producing High Fidelity Systems," *Computer Design,* Vol. 15, No. 1, p. 63; Jan., 1976.

J.C. Tenny, "Sound Generation by Means of a Digital Computer," *JMT,* No. 7, p. 24; 1963.

H.M. Tremaine, *Audio Cyclopedia,* Sams, N.Y.; 1969.

D.E. Walker, "The SRI Speech Understanding System," *IEEE,* Vol. ASSP-23, No. 5, Oct., 1975.

L. Wedin & G. Goude, "Dimensional Analysis of the Perception of Instrumental Timbre," *Scand. Jour. Psych.,* No. 13, P. 228; 1972.

S. White, "On Mechanization of Vector Multiplication," *IEEE,* Vol. 63, No. 4; April, 1975.

T. Winograd, "Linguistics and the Computer Analysis of Tonal Harmony," *JMT,* Vol. 12, No. 1, p. 2; 1968.

J. Zingheim, "Introduction to Computer Music Techniques," *EN,* Vol. 6, No. 42, p. 1; Aug., 1974.

ABBREVIATIONS USED, AND ADDRESSES

ACM — Association for Computing Machinery

Acustica, S. Hirzel, Stuttgart 1, Birkenwaldstr, 44, Postfach 347, Germany.

AES — Audio Engineering Society

ASSP — *IEEE* Transactions on Acoustics Speech and Signal Processing

Communications of the ACM
1133 Avenue of the Americas, NYC 10036

Computer Design
Box A, Winchester MA 01890

Computer Journal
British Computer Society
29 Portland Place
London, England W1N 4AP

Computers and Automation [now called *Computers and People*]
Berkeley Enterprises, Inc.
815 Washington St, Newtonville MA 02160

Datamation
Technical Publishing Co.
1301 South Grove Ave., Barrington IL 60010

ED — *Electronic Design*
50 Essex St., Rochell Park NJ 07662

EDN — *EDN Magazine*
270 St. Paul St., Denver CO 80206

EN — *Electronotes*
203 Snyder Hill Rd., Ithaca NY 14850

IEEE — *Institute of Electrical and Electronics Engineers*
345 E 47 St., NYC 10017

JAES — *Journal of the Audio Engineering Society*
60 E 42 St
Lincoln Blg, Room 929
NYC 10017

JASA — *Journal of the Acoustical Society of America*
American Institute of Physics
335 E 45 St., NYC 10017

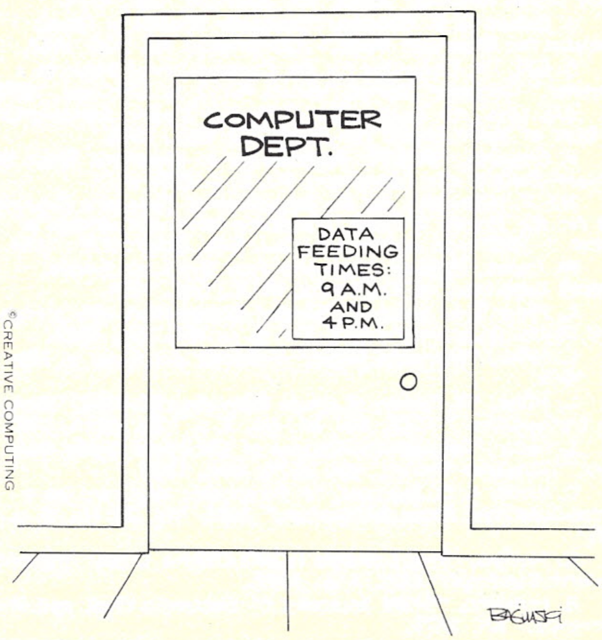
JMT — *Journal of Music Theory*
Yale School of Music
Yale University
New Haven CT 06520

NW — *NUMUS West*
Box 135, Mercer Island WA 98040

Perspectives of New Music
Box 231, Princeton NJ 08540

Science
American Association for the Advancement of Science
1515 Massachusetts Ave. N.W.
Washington DC 20005

Scientific American
415 Madison Ave., NYC 10017



A Computer Music System for Every University: The Dartmouth College Example

by
Sidney Alonso*
Jon H. Appleton
Cameron Jones

It has become clear in recent years that the future of musical performance, composition, distribution and instruction is increasingly dependent on technology. These developments have expressed themselves at Dartmouth College where there has been a decade of activity in the composition and performance of electronic music, as well as a tradition of computer-assisted instruction in many areas, recently including music.

The last few years have also witnessed startling and rapid advances in digital electronics technology enabling computer methods to be applied inexpensively to many phenomena of the "continuous world" which were formerly the province of analog technology. Research in this area has been carried on for some time in the Computer Hardware Laboratory of the Thayer School of Engineering and at the Kiewit Computation Center at Dartmouth College.

The history of the relationship between music and technology has been well documented, most recently in an article entitled "Computers and Future Music" by Mathews, Moore and Risset *(Science, Vol. 183, No. 4122, 1974)*. This article describes the development of a digital synthesizer that was built and is being used at Dartmouth in the instruction, composition and performance of music. The actual design and construction, as well as the use of the system, is the result of the convergence of curricular needs and was accomplished through collaboration of the music department and the engineering school.

Analog synthesizers had been used almost exclusively in the composition and performance of electronic music and the situation at Dartmouth was no different. However, because there had been significant activity in this musical genre at Dartmouth, it was natural that interest in digital synthesis should manifest itself because of the greater control it affords the composer. At the same time it was recognized that the traditional approach to digital synthesis (Music V, Music 360, etc.) did not satisfy the compositional expectations concerning ease of use and low cost. The procedures used in conventional digital synthesis programs are at best cumbersome for most composers, and impose certain "stylistic" limitations. The greatest obstacle, however, which has made live performance impossible, has been the "turn around time" required before the composer can hear what he has specified.

Within the more traditional offerings of the music department there have been theory courses where part of the objective is to train the "ears" of students in such skills as interval, scale and chord recognition. Because this ear training is best accomplished on an individual basis, we had been using one of a number of methods which have musical examples on tape with an accompanying workbook. The difficulty here is that the method does not give any feedback nor does it proceed at a rate appropriate to an individual student. Again, it was natural that we should become interested in computer-assisted instruction and our thoughts turned to digital synthesis in this connection.

A third, and perhaps more innovative application grew out of our interest in having beginning music students learn about the subject by actually composing music. We recognized that this could only be done if they could hear their work "performed" and if the musical framework were simplified so that they could concentrate on only one or two aspects of composition at a time, i.e., sound material, form, etc. With the use of a digital synthesizer, a simple language, and several carefully constructed exercises it has been

* Dartmouth College, Hanover, NH 03755.

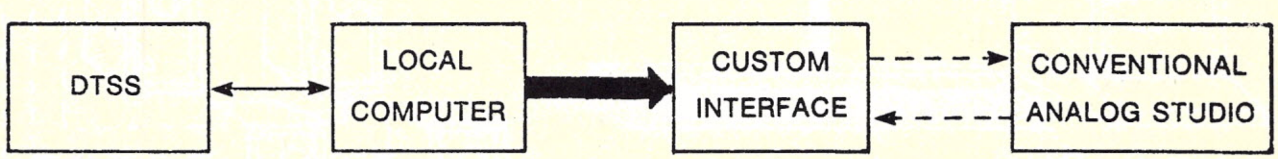
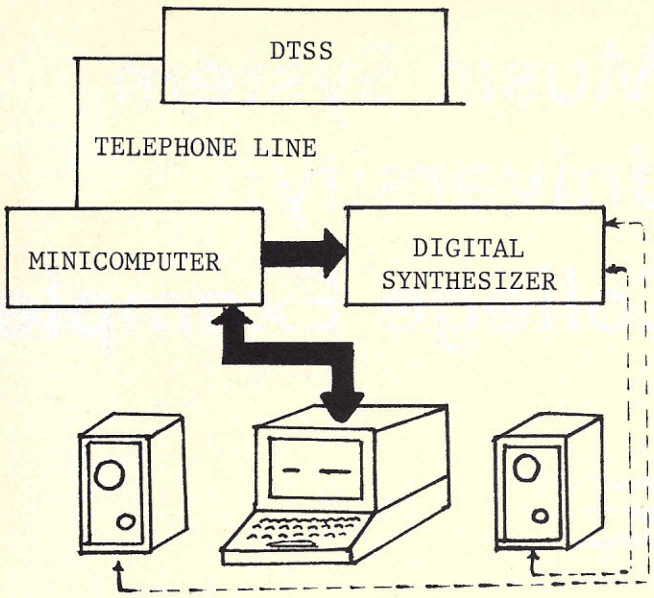


Fig. 1. Computer control of an analog studio.

Fig. 2. Early music instruction configuration at Dartmouth College.



possible to introduce musical composition to beginning students. Whether in fact students learn more about how to listen to music than they would in a conventional introductory course is impossible to state at this time.

The various musical needs described above arose at a time when our Computer Hardware Laboratory was experimenting with the design of various devices which would interface with the large Dartmouth time-sharing system, DTSS. Using the idea that any device that can be made to mimic a terminal can use DTSS as a "host" computer, several interesting systems had been developed. In one, DTSS was used as a large memory for a "smart terminal"; in another, it was used as the arithmetic and logic processor for a small, student-built computer. At this time the question arose as to whether the DTSS could somehow be used to control the conventional analog sound production and modification equipment of Dartmouth's Bregman Electronic Music Studio. The musicians and engineers soon realized that more "real-time" computing power would be needed for this task than could be reasonably expected from a general-purpose time-sharing system. It was easy to imagine using the time-shared host computer as the storage medium for rather massive files of musical data, but the control of the synthesizing equipment itself called for a local computer (Fig. 1).

When the interface between the small computer and the existing analog synthesis equipment was studied it became clear that the translation between the discrete or digital "language" of the computer and the continuous analog "language" of the synthesizer would involve equipment of a complexity and expense possibly equal to the small computer itself. For example, consider the problem of frequency control. Even though one might employ high precision, and high cost, digital-to-analog converters to provide the control voltages for analog oscillators, conventional synthesizer oscillators, and for that matter other associated synthesizer modules, are not high precision equipment and are subject to both long and short term drift.

On the other hand, current digital frequency control methods provide outputs which are numerically related to a single master oscillator whose stability can be made extremely high; for example by means of a quartz crystal resonator. Since frequency ("pitch" to musicians) is one of the fundamental parameters of music, an entirely digital system seemed the only logical course.

The authors began to formally collaborate in 1972 with financial assistance secured by Dartmouth President John G. Kemeny, toward the realization of a system which would

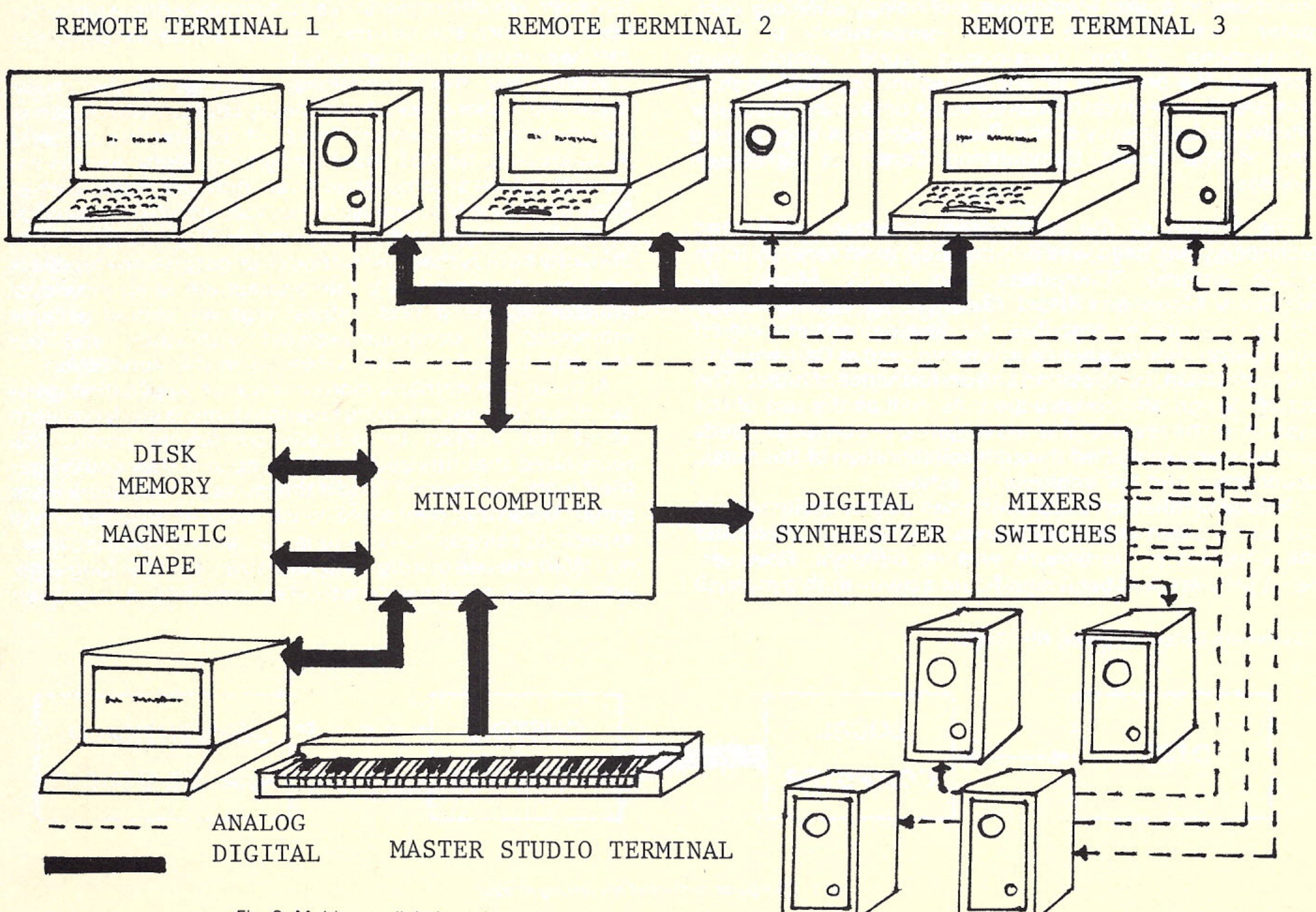


Fig. 3. Multi-user digital music system.

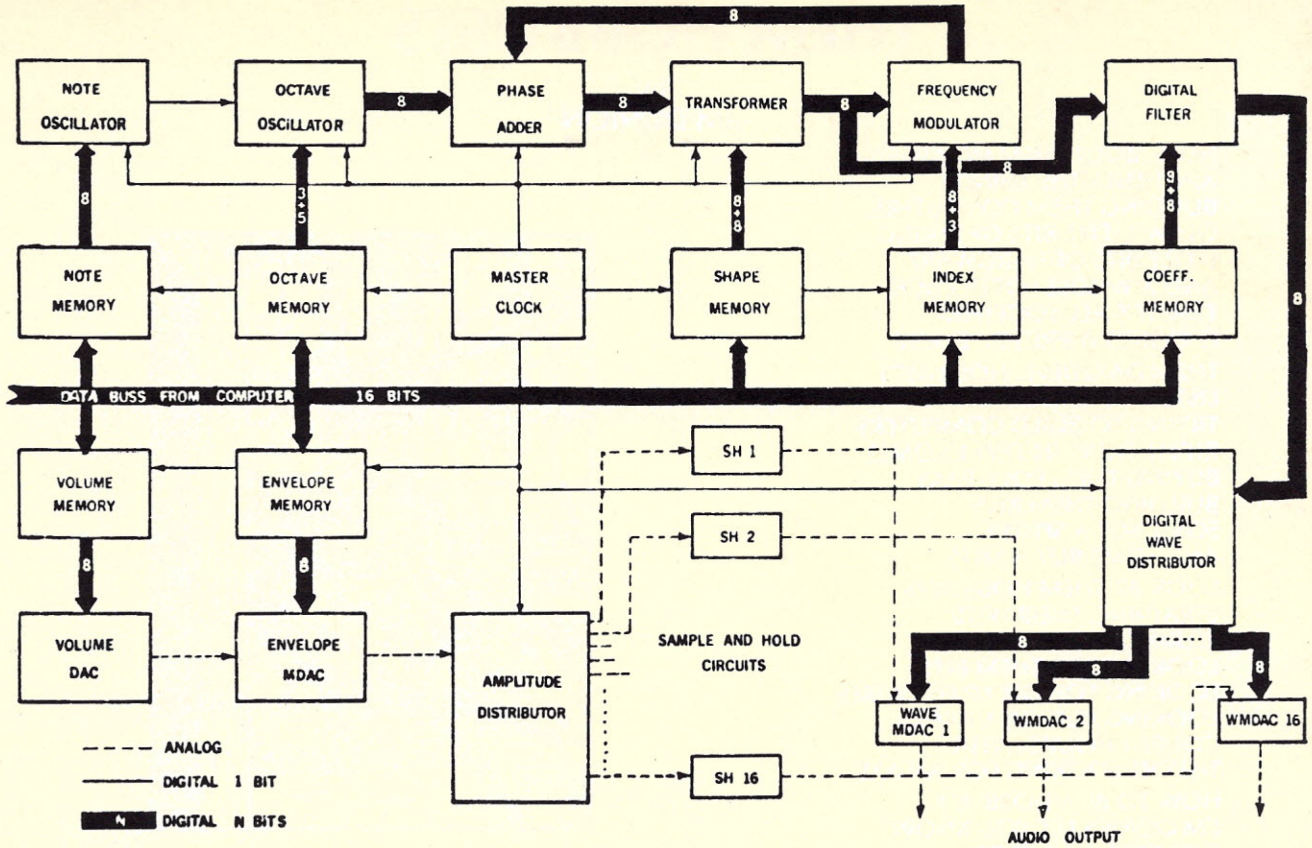


Fig. 4. Block diagram of digital synthesizer.

meet the several musical-instructional requirements outlined above.

A computer terminal was installed in the music department with sound output from a digital synthesizer and verbal input/output from a CRT/keyboard terminal. It was locally controlled by a 16-bit minicomputer to achieve short response times. The music and verbal data were stored in DTSS and accessed over a 300 Baud telephone link. To the student the terminal appeared similar in operation to a normal time-sharing terminal. The synthesizer required minimal coding to produce a variety of musical timbres with accurate intonation and reproducibility. Each of eight identical channels provided a high resolution frequency source driven from a common 11.5 MHz clock. Channels produced ramp or triangle waves with six bits of shape resolution, and a variety of rectangular waves derived from square waves. The ramp/triangle and rectangular waves were digitally amplitude controlled with six bits of resolution each. These volume controls permitted generation of a variety of envelope functions, including those with time-variable harmonic content. Channel outputs were filtered, weighted, and combined before transmission to a conventional stereo amplifier system (Fig. 2).

This system was used in connection with four different music courses during 1973-1974: three were music theory courses in which students developed traditional musical skills dealing with melodic, harmonic and rhythmic perception, and one course presented exercise in elementary composition.

Special software was written for both the large time-sharing system and the local minicomputer. The program running on DTSS retrieved data stored there which was then formatted and sent over the telephone link to the minicomputer. This data included the music theory exercises in the form of questions, answers and machine-coded descriptions of musical passages. The minicomputer, using its real-time clock, translated the music data into synthesizer instructions. For example, a sixteen-bit number set the frequency of one of the eight oscillators.

Once a student signed on the system and selected the appropriate exercise, a question appeared on the CRT screen explaining that the music about to be played represented a harmonic, melodic or rhythmic concept. After hearing the music, the student typed one of several possible responses and was immediately informed if the response was the correct one. Individual as well as sets of questions were instantly repeatable creating an infinitely patient learning environment.

Beginning composition exercises allowed the student to arrange strings of pre-defined sound objects to create a short musical composition. These compositions could be edited and stored in DTSS to be accessed and played in much the same manner as the theory exercises.

During the first year of operation, approximately three hundred students used the system on a regular basis and showed considerable enthusiasm for this mode of instruction. The system was reliable but there were simply not enough hours in a week to adequately meet the student demand. We also realized that the synthesizer itself could be improved with respect to tone quality and timbral resources. Other colleges and universities expressed interest in having our system since the instructional and compositional problems it addresses are common to most educational institutions. Our link to DTSS prohibited export and we began to think of other alternatives.

In 1973 we received the support of the Sloan Foundation as part of a large grant to Dartmouth College to improve instructional technology and services. Part of the requirement for this grant was the charge to create an exportable, low cost system (c. $20,000.) that could serve the beginning

THEM HOBBYISTS

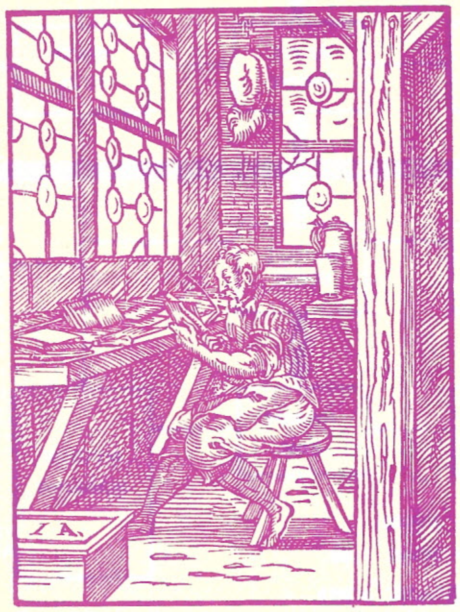
by
JIM DUNION

HOW 'BOUT THEM HOBBYISTS,
AIN'T THEY THE THING?
BUILDING THEM COMPUTERS,
OUTA' LITTLE BITS OF STRING,
LITTLE BITS OF HARDWARE
AND A WHOLE LOTTA LUCK,
TIME IT'S ALL TOGETHER,
IT COSTS A PRETTY BUCK!

THEM DAREDEVIL HOBBYISTS
LISTEN TO 'EM MOAN
TRYING TO BUILD COMPUTERS
RIGHT THERE IN THEM HOMES
BUYING THEM POLY-PAKS
BUILDING THEM KITS
SOME BUY A SPHERE
AND SOME BUY A MITS.

LOOK AT THEM HOBBYISTS
STRAINING THEIR WITS
PEERING IN THE INNARDS
LOOKING FOR THEM BITS.
LOOKING FOR THE LOOSE WIRES
LOOKING FOR THE CRACKS
COUPLA HOURS LATER
THEY'RE LOOKING FOR AN AXE

HOW TO BE A HOBBYIST
I'M GONNA LET YOU KNOW
GIT YERSELF A MICRO
AND WATCH IT START TO GROW.



student as well as the advanced composer. We decided on the system configuration shown in Fig. 3.

Four students can run exercises simultaneously but the entire resources of the system are available to a composer from one master terminal. Users communicate with the system by means of a silent CRT terminal, piano keyboard and other input devices.

The new synthesizer consists of sixteen independent channels or "voices"; thus each of the four student terminals may independently reproduce four-part musical passages or exercises. Each channel can produce a wave of arbitrary harmonic content up to the 16 KHz limit of normal hearing. For the musician this means rich, complex "natural" timbres instead of the arbitrary ones available previously.

The unnaturalness of synthesized sounds that has disturbed many musicians can be attributed to the fact that, unlike real instruments, easily synthesized waves do not have time varying harmonic content. The work of John Chowning in recent years has created a simple but effective means for bringing the harmonic content of synthesized waves under greater control. By means of Frequency Modulation, where many harmonics can be controllably produced from a pair of sine waves, one is able to realize highly desirable timbral content with computational parsimony. The real-time solution of the FM algorithm was therefore included in the new synthesizer design.

Naturalness is also a result of subtle temporal control of amplitude. In the new design each channel has an amplitude which is the product of two independent 8 bit numbers, "Volume" and "Envelope" in Fig. 4.

Figure 4 also shows the highly time-multiplexed use of the hardware modules. Each functional block performs its operation in time succession on each of the sixteen channels. Because of the extremely high speed of operation of modern digital circuitry, it is possible to perform many independent calculations with the same physical hardware, and by sequencing these calculations have them appear to occur simultaneously. Thus one time-multiplexed digital oscillator does the work of sixteen analog oscillators. Because each of the sixteen channels can produce a highly complex sound, this synthesizer represents a powerful new medium to the composer of electronic music working at the master terminal.

When the instructional mode is used in which four oscillators are assigned to each of four student terminals, it becomes necessary to time-share the minicomputer as well as the synthesizer. A general purpose time-sharing system was implemented on a 24k, 16 bit minicomputer with a fixed-head disk and a tape drive. Users may sign on to this system and create, edit and save programs which also may be compositions. The user interface superficially resembles the familiar DTSS. Theory and composition exercises are similar to those used on our earlier system except that the quality of the sound is greater allowing for a more familiar acoustical environment for theory exercises and more interesting sound material for the beginning composer. It should be stated again that the new system is serving four students during the time that the earlier system served one.

To aid in the software development of special purpose music playing, and editing programs, a subset of PK-1 was implemented on the mini-time-sharing system. This higher level language allows composers to structure the system in such a way as to reflect their individual stylistic approach to musical composition without requiring an intimate knowledge of the computer or assembly language programming.

We believe that this system represents a step forward in making digital synthesis available to composers and further extends its use to computer-assisted instruction in music at low cost. Computing machines as an extension of musical thought are one of the most exciting cultural-technological developments of our time. ■

The Floating Point Solution

David Ahl
CREATIVE COMPUTING

Dear Mr. Ahl:

Nicholas Meyer discovered the previously unknown Sherlock Holmes adventure, *The Seven-Per-Cent Solution* and electrified Holmes's present day followers with the establishment of a link between Holmes and Freud. I have just discovered a link between Holmes and several of the pioneer thinkers in computing, an exciting tale which Watson appropriately dubbed *The Floating Point Solution*. Certainly there are descriptions of Holmes in many of the previously published tales which suggest how close his mentality was to that of the typical programmer/analyst. Nothing, though, even approaches the truth embodied in *The Floating Point Solution*.

Unfortunately, the manuscript itself is not yet in a publishable form and so I can not really submit it to you for serialization in *Creative Computing*. At this time, the most I can do is allow you to publish a few of the Sidney Paget illustrations which had been prepared for this tale but were never published. The captions will convey a bit of the drama of the tale itself but will not, I trust, give the story away completely.

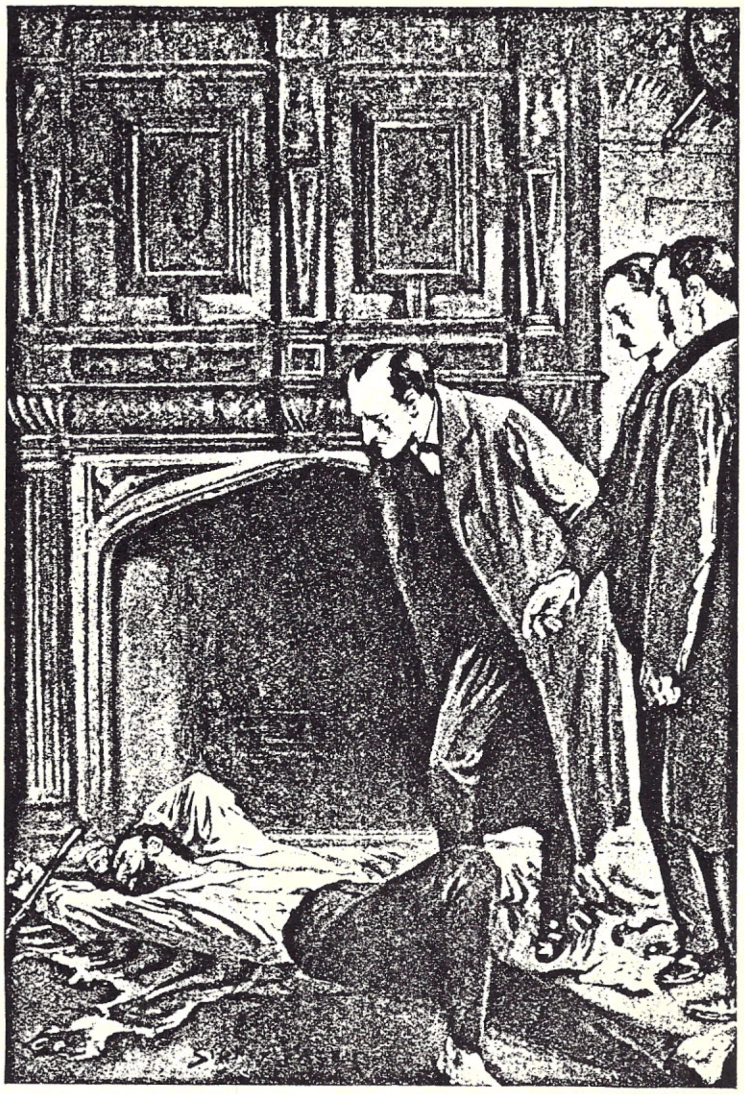
You may expect to hear from me again.

Sincerely yours,

Robert P. Taylor, Esq.
Kings College, Budo

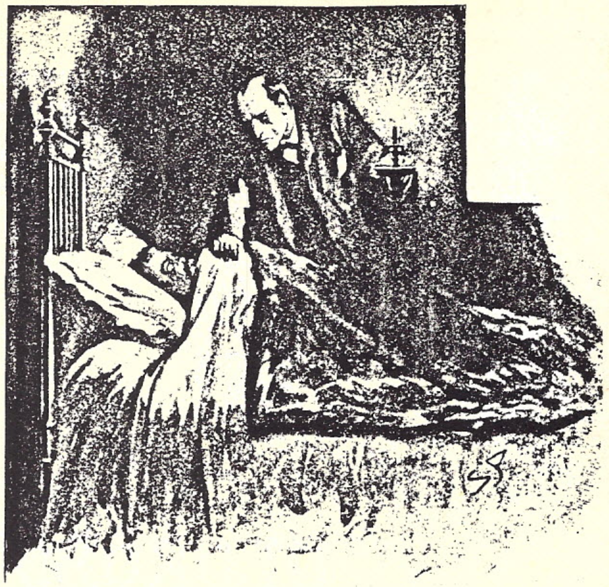
"Delete it!"

"....it <u>has</u> to be a hardware error..."



"Yes," mused Holmes, pondering the unfortunate operator's body, "one never knows in advance how the disgruntled user may react......."

"My guess, Watson, is that it's some sort of artificial brain..."



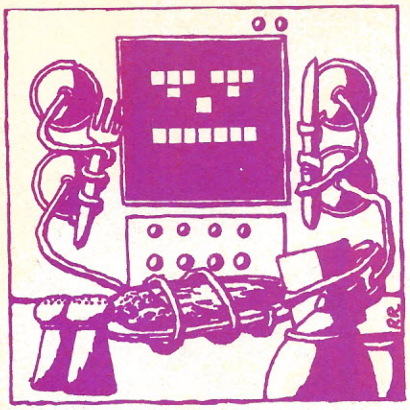
"Hurry, Watson!" cried Holmes. "This is no accidental power loss."

"It must be secret code," thought Holmes, "....it certainly doesn't make sense as a users manual...."

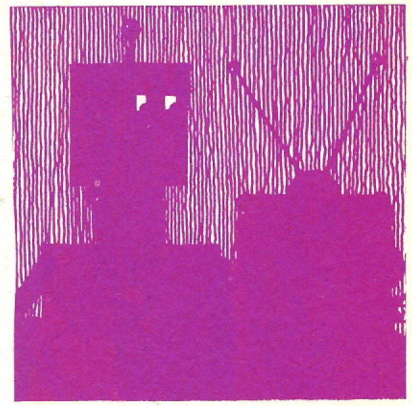
"Come, come," said Holmes, kindly, "at least you still have the listings."

GLOROBOTS

Gloria Maxson

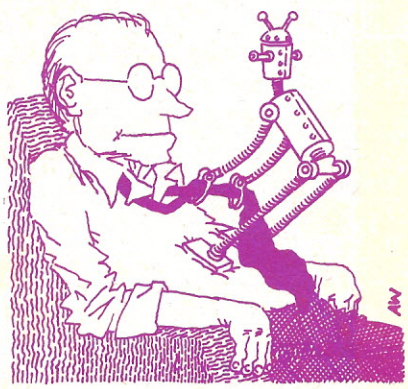


A computer invited to sup,
fumbled with platter and cup,
with fork and with knife,
and, asked about life,
mumbled, "It doesn't add up!"



A robot of amorous renown
broke the whole powerplant down
by starting to pet
with an old TV set,
which totally blacked out the town.

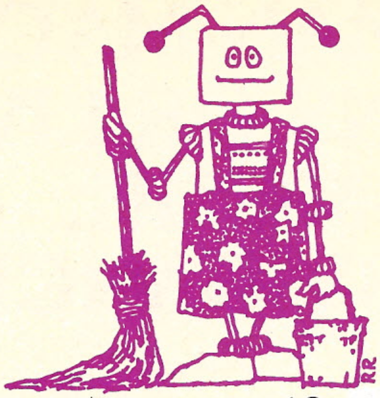
There'll be computers of size,
and some they will miniaturize
to do the small tasks
that everyone asks,
from scratching, to knotting of ties.



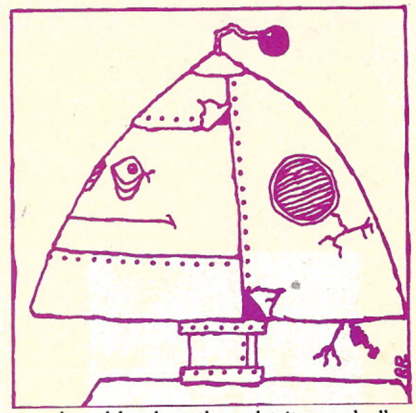
A new synthesizer named Moog,
and an old-fashioned organ named Oog,
had a concert to test
which one played best,
and are having it yet, Moog and Oog.

I heard an old robot explain:
"We were evolved to be sane,
with tolerant views,
so men need not use
the other nine-tenths of their brain."

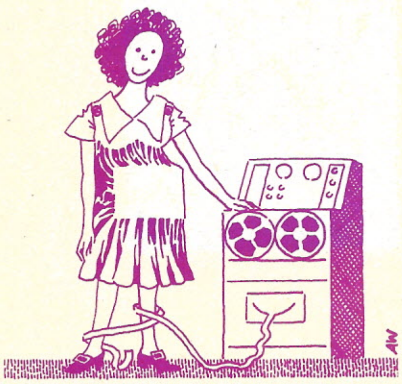
A robot with lofty inflection
read Stein in the poetry section,
but read it, "Arose
is arose is arose,"
and thought it concerned resurrection.



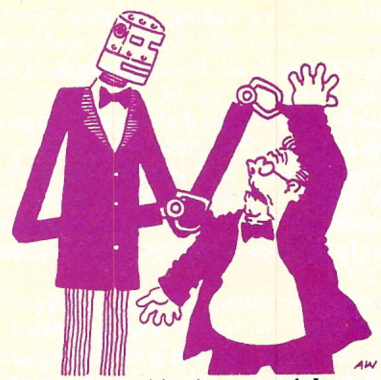
A robot domestic named Grace
would work any time, any place,
and never relent,
because she was bent
on expanding the customer base.



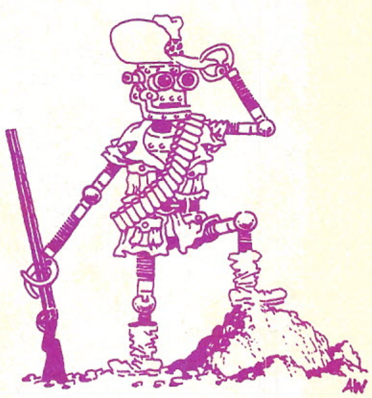
An old robot thought it was hell
that he was unable to tell
how feeble he got,
since it was his lot
to be terminal even when well.



A computer designed to compute
a couple's rapport or dispute,
typed "Yes!" unashamed
when the young lady named
it in her paternity suit.

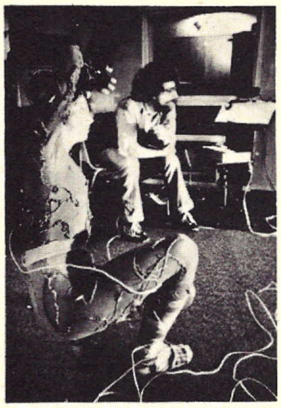


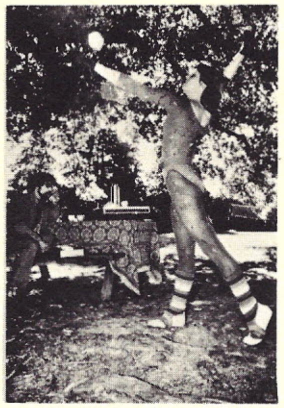
A robotical butler named Jeeves
was given no further reprieves
after asking a guest
for his coat, hat — and vest,
shoes and socks, shorts, and shirtsleeves.



A robot in tracking had fun
in a way calculated to stun:
he wore hunting suits,
bold hats, and big boots,
and carried an elephant gun.

INTERACTIVE WOMAN-MACHINE IMPROVISATIONS OR LIVE COMPUTER-MUSIC, PERFORMED BY DANCE





Debra Loewen is wearing a gravity-sensor costume that is monitored by a computer and correlated instantaneously with sound from a synthesizer. Depending on the user's program, Debra could be performing a new musical instrument, executing a dance piece that determines its own unique music score in real time, or experimenting with bio-feedback through physical movement and sound.

The costume has 64 mercury switches (as digital bits) multiplexed to single data lines for shipment (by cable or radio) to an interface. The system hardware — designed and built by Thomas Noggle and Joseph Pinzarrone in Urbana, Illinois, 1972-1975 — consists of a READ/WRITE bus structure between a P.D.P. 11/10 and a programmable bank of analog sound-snythesizer modules (i.e. oscillators, filters, amplifiers etc.).

System software, coded by Elven T. Riley and T. Rust, allows user control of analog device interconnection and data as well as costume information monitoring and masking and — can be used to create real-time interaction between the two.

The system has been featured chiefly in live performance from coast to coast, with Debra as interfacee in a myriad of theatrical situations including composer Pinzarrone's theatre work *Hunger Artist.* Currently, a lecture-demonstration is available entitled "Interactive Woman-Machine Improvisations." Expansion of the system, of course, remains proportional to financial support.

Debra is a choreographer-dancer on the faculty of the University of Delaware. Thomas is an engineer with High Energy Physics at the University of Illinois. Joe P. (in photo background) is a composer and co-director of the Mid-Atlantic Modern Music Institute in Wilmington, Delaware.

Electronic and Computer Music

Reprinted below is the list of currently-available electronic and computer music tapes and records as listed in the January 1977 *Schwann-1 Catalog.*

Key:	
Q	Quad record
▲	8-track tape
Q▲	Quad-8 tape
•	Cassette Tape

Label abbreviations:

Adv.	Advance
Ang.	Angel
At.	Atlantic
Can.	Candide
Col.	Columbia
DG	Deutsche Grammophon
Ev.	Everest
Fin.	Finnadar
Folk.	Folkways
GC	Golden Crest
Lyr.	Lyrichord
Main.	Mainstream
Mer.	Mercury
None	Nonesuch
Odys.	Odyseey
Ser.	Serenus
Van.	Vanguard
West.	Westminster

Listings below give complete contents when the entire record contains electronic music. A reference to the listing under the composer's name in the Composer Section is made when only a part of the contents of the record falls into this category. Performers' names are in italics.

Adv. **S-8—MAXFIELD:** Pastoral Symphony (1960); Bacchanale (1963); Piano Concert For David Tudor (1961); Amazing Grace (1960)
Adv. **16—BUDD:** Oak of Golden Dreams (1970); Coeur D'Orr (1969)
Ang. **S-36042;** ●4XS-36042—**WHITE:** Short Circuits Moog synthesizer & other electronics
At. **S-1576—MIMAROGLU:** Sing Me a Song of Songmy—Hubbard, Reciters, Chorus, String Orch., Organ, Mimaroglu, Synthesized & Processed Sounds
Can. **31001—STOCKHAUSEN:** Prozession for Tamtam, Viola Elektronium, Piano, Filters & Potentiometers (1967)
Can. **31025—SCHAEFFER:** Objets liés; **MACHE:** Terre de feu, 2nd version; **PHILIPPOT:** Etude III; **BAYLE:** Oiseau Chanteur; **FERRARI:** Tête et queue du dragon; **MALEC:** Dahovi; **PARMEGLIANI:** Danse
Capra **1201**—*See Duckworth, Strange, Wilson*
Col. **MS-6146**—*See Varèse*
Col. **MS-6362**—*See Varèse*
Col. **MS-6566—AREL:** Stereo Electronic Music No. 1 (1964); **BABBITT:** Composition for Synthesizer (1964); **DAVIDOVSKY:** Electronic Study No. 1 (1964); **EL-DABH:** Leiyla and the Poet (1962); **LUENING:** Gargoyles (1962); **USSACHEVSKY:** Creation—Prologue (1962)—*Col.-Princeton Elec.*
Col. **MS-7051—BABBITT:** Ensembles for Synthesizer; **CAGE:** Variations II (1961)—*Tudor;* **POUSSEUR:** Trois visages de Liège (1961)
Col. **MS-7139**—*See Brown, Bussotti, Cage*
Col. **MS-7194; MQ-31018(Q);** ▲18-11-0092; **Q▲MAQ-31018;** ●16-11-0092—**SWITCHED-ON BACH—J. S. BACH:** Brandenburg Con. No. 3; Jesu, Joy of Man's Desiring; etc. (11-68)—*Carlos, MOOG*
Col. **MS-7286;**▲18-11-0144;●16-11-0144—**CARLOS:** Well-tempered Synthesizer
Col. **MS-7355—STOCKHAUSEN:** Mikrophonie I for Tamtam, 2 Microphones, 2 Filters & Potentiometers (1964)—*Aloys Kontarsky, Alings, Fritsch, Bojé, Stockhausen, Davies, Spek;* Mikrophonie II for Choir, Hammond Organ & Ring Modulators (1965)—*Alfons Kontarsky, Fritsch, Schernus*
Col. **M-30383;**▲MA-30383;●MT-30383—**KAZDIN & SHEPARD:** Everything You Always Wanted to Hear On the Moog (9-71)
Col. **M-30683; Q—SUBOTNICK:** Sidewinder—Subotnick
2-Col. **PG-31234;**▲GA-31234;●GT-31234—**CARLOS:** Sonic Seasonings (1972)
Col. **KC-31480—CARLOS:** Clockwork Orange
Col. **XM-32088;** ▲XMA-32088; ●XMT-32088—**CARLOS:** By Request
Col. **KM-32659;** ▲KMA-32659; ●KMT-32659—**Bach:** Suite No. 2 in b; Sheep May Safely Graze; Brandenburg Con. No. 5; etc.—*Carlos, MOOG*
Col **M-32741; Q—SUBOTNICK: 4 Butterflies**
Corn. U. **1**—*See Bielawa*
Corn. U. **7**—*See Borden*
CRI **S-204**—*See Davidovsky*
CRI **S-219**—*See Luening*
CRI **S-227—LUENING & USSACHEVSKY:** Concerted Piece for Tape Recorder & Orchestra—*Col.-Princeton Electronic Music Ctr.;* **POWELL:** Events, M. (1963); Improvisation (1963); Second Electronic Setting (1962); 2 Prayer Settings—*Wilson, Schwartz, Davenny; Bressler, Kaplan, Tarack, Lynch, Kouguell;* **USSACHEVSKY:** Of Wood & Brass; Wireless Fantasy—*Col.-Princeton Elec. Music Ctr.*
CRI **S-255**—*See Druckman*
2-CRI **268—Varèse:** Déserts (1954-61); **LUENING:** In the Beginning (1956); **BABBITT;** Vision & Prayer (1961); **USSACHEVSKY:** Computer Piece No. 1 (1968); 2 Sketches (1971); **SMILEY:** kalyosa (1970); **SHIELDS:** Transformation of Ani (1970); **DAVIDOVSKY:** Synchronisms No. 5 (1969)—*Col.-Princeton Elec. Music Ctr.*
CRI **S-296—EATON:** Mass—*Hirayama, White, Syn.;* Blind Man's Cry—*Hirayama, Syn.;* Concert Music for Solo Clarinet—*Smith*
CRI **S-310—HILLER & BAKER:** Computer Cantata—*U. of Ill. Contemp Ch. Players, Hamm,* **MELBY:** 91 Plus 5—*Contemp. Brass Qn*
CRI **S-328—RANDALL:** Music for the Film "Eakins"—*Princeton U. Comp. Ctr;* **CEELY:** Elegia—*RAI Studio Fonologia; Mitsyn Music ;* **BEEP; DEL MONACO:** Electronic Study No. 2 (1970); Metagrama—*Sanoja, Col.-Princeton Music Ctr*
CRI **S-335—P. McLEAN:** Dance of Dawn (1974);**B. McLEAN: Spirals (1973)**—*Indiana U. South Bend Elect. Music Studio*
CRI **S-348—DODGE:** In Celebration; Speech Songs; The Story of Our Lives
Deram **18066—SATIE:** Electronic Spirit of Erik Satie—**Moog** Synthesizer,*Camarata, Contemp. Ch. Orch.*
Desto **6466—LUENING:** Fantasy In Space; Invention On 12 Notes; Legend; Low Speed; Lyric Scene; Moonflight; **LUENING & USSACHEVSKY:** Incantation; **USSACHEVSKY:** Sonic Contours
4-Desto **6474/7**—*See Tanenbaum*
Desto **7130;** ●47130—**TANENBAUM:** Movements; Contrasts; Blue Fantasy; For the "Bird"—ARP Synthesizer
DG **138811—STOCKHAUSEN:** Gesang der Jünglinge (1955/6):Kontakte (1959-60)
2-DG **2707039—STOCKHAUSEN:** Hymnen: Anthems for Electronic & Concrete Sounds
Ev. **3132—CAGE:** Variations IV (excerpts) (1965)
Ev. **3230—CAGE:** Var. IV, Vol. 2—*Cage, Tudor*
Fin. **9001—MIMAROGLU:** Wings of the Delirious Demon & other electronic works
Fin. **9002—DUBUFFET:** Musical Experiences
Fin. **9003—MIMAROGLU:** Music for Jean Dubuffet's Coucou Bazaar
Fin. **9005(Q); Q▲—SALZMAN:** Helix—Nosp*QUOG Music/ Theatre Ens.;* Wiretap—*Nagrin;* Larynx Music—*Ross, Silverman;* Queens Collage
Fin. **9010(Q); ▲—BABBITT:** Ensembles for Synthesizer; **SMILEY:** Eclipse; **SHIELDS:** Farewell to a Hill; **USSACHEVSKY:** Piece for Tape Recorder; **DAVIDOVSKY:** Electronic Study No. 3; **AREL:** Stereo Electronic Music No. 2
Folk. **6301—HIGHLIGHTS OF VORTEX—JACOBS:** Chan (1956); Electronic Kabuki Mambo (1955); Logos (1956); Rhythm Study #8 (1957); **LONGFELLOW:** Notes On the History Of the World (1959?); 350-2 (1959?); **LOUGHBOROUGH:** For the Big Horn (1957); **TALCOTT:** Loop Number 3 (1957); Trilogy (1957)
Folk. **33436—ELECTRONIC MUSIC—GRAUER:** Inferno; **IVEY:** Pinball (1965); **ROBB:** Collage (1964); **LeCAINE:** Dripsody (1955); **AITKIN:WALTER-OLNICK-SCHAEFFER:** Summer Idyl Noesis (1962); **M. SCHAEFFER:** Dance R 43 (1961); **STEPHEN:** Fireworks; Orgasmic Opus
Folk, **33435—ROBB:** Rhythmania & Other Electronic Musical Compositions
Folk, **33437—APPLETON:** World Music Theater
Folk. **33438—ROBB:** From Razor Blades to Moog
Folk. **33440—NELHYBEL:** Outer Space Music
Folk. **33441—MIMAROGLU:** Tract (1972-74)—*Tuly Sand, American Center For Students & Artists (Paris), Columbia-Princeton Electronic Music Center*
Folk. **33869—COPE:** Navajo Dedications
Folk. **33904—AIN:** Used To Call Me Sadness—*Matsua;* **CHADABE:** Echoes—*Williams;* **McMILLAN:** Whale I; Carrefours; **MUMMA:** Cybersonic Cantilevers; **USSACHEVSKY:** Conflict
Fin. **9010(Q); Q▲—BABBIT:** Ensembles for Synthesizer; **SMILEY:** Eclipse; **SHIELDS:** Farewell to a Hill; **USSACHEVSKY:** Piece for Tape Recorder; **DAVIDOVSKY:** Electronic Study No 3; **AREL:** Stereo Electronic Music No. 2
GC **S-4085**—*See MacInnis, Trythall*
GC **S-4092—KNIGHT:** After Guernica (1969); Refractions for Clarinet & Tape (1962)—*Sweetkind;* Origin of Prophecy (1964); Luminescences (1967)
Lyr. **7210—SAHL:** Tropes on the Salve Regina
Main. **5002—BRYANT/CURRAN/RZEWSKI/TEITELBAUM/ VANDOR :** Live Electronic Music Improvised—MEV. Rome; **CARDEW/GARE/HOBBS/PREVOST/ROWE:** Live Electronic Music Improvised—AMM. London
Main. **5003**—*See Kagel*
Main. **5004**—*See Berio*
Main. **5005**—*See Cage*
Main. **5010—ASHLEY:** Purposeful Lady Slow Afternoon; **BEHRMAN:** Runthrough; **LUCIER:** Vespers (1968); **MUMMA:** Hornpipe (1967)—Sonic Arts Union
Main. **5015**—*See Cage*
MCA **2220—SANTO:** John Santo Plays Bach
Mer. **80000;** ▲8-80000; ●4-80000—**GLEESON:** Beyond the Sun (Holst:Planets)
None. **71174—SUBOTNICK:** Silver Apples
None. **71198—RUDIN:** Tragoedia
None. **71199—GABURO:** Antiphonie III (Pearl-White Moments); Antiphonie IV (Poised); Exit Music I:The Wasting of Lucrecetzia; Exit Music II:Fat Millie's Lament—*New Music Choral Ens., Univ. of Ill. Contemp. Ch. Players*
None. **71208—SUBOTNICK:** Wild Bull
None. **71223—ERB:** Reconnaissance (1967)—*Erb, Douglas, Forbes, Grierson, Watson, Thomas, Stein;* In No Strange Land (1968)—*Dempster, B. Turetzky*
None. **71224**—*See Cage & Hiller*
None. **71225—WUORINEN:** Time's Encomium, for synthesizer (1968-9)
None. **71245—RANDALL:** Quartets in Pairs (1964); Quartersines (1969); Mudgett (monologues by a mass murderer) (1965)—*Kessler;* **DODGE:** Changes; **VERCOE:** Synthesism—*Columbia-Princeton U's Computer Centers*
None. **71246—XENAKIS:** Bohor I (1962); Orient-Occident III (1959-60); Diamorphoses II (1958); Concret P-H II (1958)—*ORTF, Paris, Groupe recherches musicales*
None. **71250—DODGE:** Earth's Magnetic Field—*Boller, Frederick, Ungar, Columbia U. Computer Center*
None. **71289—KORTE:** remembrances, for Flute & Tape;**DAVIDOVSKY:** Synchronisms No. 1, for Flute & Tape; **KUPFERMAN:** Superflute, for Flute & Tape—*Baron*
2-None. **HC-73018**($11.88, w. book)—**NONESUCH GUIDE TO ELECTRONIC MUSIC**
Odys. **32160156**—*See Cage, Ichiyanagi, Lucier*
Odys. **32160158**—*See Mumma*
Odys. **32160160—MAXFIELD:** Night Music (1960); **OLIVEROS:** I of IV (1966); **REICH:** Come Out
Odys. **Y-34139—CANN:** Bonnylee (1972); **GRESSEL: Points In Time (1974) KREIGER:** Short Piece (1974); **LANSKY:** mild und Leise (1973-1974); **SEMEGEN:** Electronic Composition No. 1 (1971-1972); **WRIGHT:** ElectronicCompostion No. 2 (1973); **ZUR:** Chants, for Magnetic Tape (1974)
Odys. **Y-34158—SUBOTNICK:** Until Spring
Op. One **7**—*See Schubel*
Op. One **17—McCLELLAN:** Distant Voices (1971); **CHADABE;** Ideas of Movement at Bolton Landing (1971)
Opus One**24—McCLELLAN:** Genesis (1974); Interruptions (1971)
Orion **7021;**●CAS-70211—**SWICKARD:** Sermons of Saint Francis, for narrator and tape (Moog) (1968); Hymn of Creation for narrator and type (1969)—*DuBay;* **HELLER:** Labyrinth for Cello & Tape (1969)—*Ischar*
Orion **74142—GRAYSON:** Live Electronic Music
Orion **75192—B. McLean:** II, for Piano & Tape—*Hamilton;* The Sorcerer Revisited; Genesis
Point **101**—*See Badings*
RCA **ARL1-0488; Q;** ▲ARS1-0488; **Q▲;** ●ARK1-0488—**TOMITA:** Snowflakes Are Dancing (Music of Debussy)
RCA **ARL1-0838; Q;**▲ARS1-0838; **Q▲;** ●ARK1-0838—**TOMITA:** Pictures at an Exhibition (Mussorgsky)
RCA **ARL1-1312; Q;** ▲ARS1-1312; **▲;** ●ARK1-1312—**TOMITA:** Firebird (Music of Stravinsky, Debussy, Mussorgsky)
Ser. **12045—CUSTER:** Found Objects No. 5 for 5 Instr. & Tape—*Custer, Ens.;* No. 8, for Violin & Tape—*Kobialka;* No. 6, for Flute & Tape—*Shansky;* No. 3, for Contrabass & Tape—*Turetzky*
Van. **C-10057—RANDALL:** Lyric Variations for Violin & Computer (1966-7)—*Zukofsky;* **SAHL:** Mitzvah For the Dead, for Violin & Tape (1966-7)—*Zukofsky*
Van. **C-10069—CZAJKOWSKI:** People the Sky
Varese **81001—TAYLOR:** Lumière, for Synthesized & Concrete Sound
West. **8110;**▲F8110—**GASSMANN:** Electronics: Music to the Ballet (1961); **SALA:** Five Improvisations
West. **8129;**▲F8129—**MEYERS:** Rhythmus; Excitement; In Memoriam for Soprano & Tape; Chez dentiste; Moonflight Sound Pictures—*Hansel;* Intervals I; **MARDIROSIAN:** Fantasia for Organ & Tape—*Mardirosian;* **HEINTZ:** Fanfare & Raga for Bassoon & Tape—*Heintz; Meyers, Catholic University of America Electronic Music Laboratory*
West. **8182—Unusual Classical Synthesizer**—*Hankinson, Putney V.C.S. 3 Synthesizer*

AN INSIDER'S GUIDE TO COMPUTER MUSIC RECORDINGS

John Selleck

Computer music appears to most people as a kind of apotheosis of electronic music. It can be thought of as fitting into the electronic music category, but it has many distinct features that are not found in electronic studio music or most synthesizer music.

Computer music is not oriented toward performer usage. It is a composer's medium and even if the processes can work in real-time, the forte of the computer is its data-processing capability as applied to all kinds of information, not just its ability to simulate a musical instrument. Real-time situations are at present more often used for preliminary organizations of musical materials, with the intent to use the more time-consuming but more flexible processes for a final result.

The story of computer music begins at Bell Telephone Laboratories with the efforts of Max Matthews and others. In the late 60's the essential characteristics of sound-synthesis programs were established and most subsequent improvements to this early work have been in the nature of user (composer) oriented modifications. Vladimir Ussachevsky, an undisputed master of the electronic medium, produced a computer piece as early as 1968 using preliminary efforts at digital sound synthesis made by Jean Claude Risset (known for his work in brass-instrument tone synthesis) and F.R. Moore, the resulting work titled: *Computer Piece No. 1.* Already we can detect certain new features of the computer medium. The agile display of complex textures, the subtle modifications in timbre, and the sophisticated rhythmic placements are all attributes of computer sound materials that are difficult, if not impossible to obtain with manually operated analog devices. The additional possibility of real-time examination of computer generated sound, allowing for more carefully considered composer decisions, is demonstrated by another of Ussachevsky's computer pieces: *Sketches for a Computer Piece.* Here he used the GROOVE program at Bell Labs, depicted in grossly simplified form in figure 3. Again the timbral changes (often in the course of a single note) are of a much higher level of complexity than that usually obtainable in an analog studio. This piece also demonstrates the synthesis of brass-like tones. There are several sequences of "pinging" sounds that were created by the use of random-number sequences (controlling rhythmic succession and amplitude), a feature also found on some synthesizers; but again the flexibility provided by the computer, i.e., the ability to program (symbolically construct) whatever kind of sound or sound sequence that one can conceive of, clearly demonstrates the power of the computer medium. Both the pieces of Ussachevsky mentioned above are available on CRI SD 268, an anthology of computer music from the Columbia-Princeton Electronic Music Center in celebration of their Tenth Anniversary.

The computer has to be programmed by humans to produce any kind of output. The crux of the matter lies in how many decisions are going to be the result of "filtered" randomness or a statistical application of order upon chaos, and how many of them are actually made by a human composer. *HPSCHD,* created in collaboration with John Cage (appearing on NONESUCH H-71224) is an example of how such music is used as material for further post-computer manipulations. The result is playfully charming and a clear example of certain aesthetic trends that think of music as a display of textures, a mosaic of bits and pieces, surfaces rather than dramas. The result of this bias is undoubtedly due to certain aspects of the electronic (including computer) mediums as is explained in the writings of Marshall McLuhan. The desire is to inter-mix and juxtapose materials that are not really discrete sequences of clearly defined elements, but are whole textures or sound-worlds in themselves.

Another of Hiller's experiments, the *Computer Cantata (1963),* (available on CRI SD 310) was an early effort and more a picture of a process than a composition as we normally think of it. After the preliminary decisions as to the probability of events happening were made, the results from the computer, of whatever quality or effectiveness, were retained as a record of the process. The output from

the computer was not synthesized sound, although some of the music was performed subsequently by means of computer synthesis. The computer merely produced a printed output of its choices of events (notes and rhythms); see figure 6. Stylistically the piece is less of a texturally and mosaically perceivable structure than *HPSCHD*. The events forced themselves upon the listener in such a way as to stress the literalness and serious consideration of each discrete happening. The music lacks a sense of depth, although the rhythm presents the impression that something might be happening at one moment that is related to some subsequent events. In theory, at least, the idea of statistically generated musical sequences has been put aside owing to recent linguistic explanations of music which stress deep structural connections as motivating the construction of a musical work; it is not merely a matter of order (redundancy) applied in varying degrees upon a random situation. Nevertheless, the use of random structures in music for textural purposes is very prevalent, and Hiller's efforts cannot be casually dismissed without at least acknowledging that his hypothesis has stimulated serious thought about the nature of musical creation.

Other uses of algorithmically generated music that may or may not use random number sequences as a basis are found in a work by Barry Vercoe, to be discussed later, and in a work by Hubert Howe that appears on a recording put out by the American Society of University Composers as a supplement to Volume 7/8 of their *PROCEEDINGS* (it can be obtained from the society free by subscribing to that issue of their journal). Howe's contributions to computer music are extensive. He was involved in many of the programs created at Princeton University and has designed a modified version of Barry Vercoe's MUSIC 360 program (or one might characterize it as a substantial variant) called MUSIC7 (for the Sigma 7 computer). His piece, *Freeze (1472)* was, if my memory serves me, generated from relatively little initial material. I.e., the "composition" of the work involved the construction of algorithms whose employment was a composer decision, but whose resulting musical displays were logical consequences. The work has a static quality (perhaps determining the title of the work?), textures emerge and recede slowly; changes in general timbral quality are of a fairly complex nature.

Also appearing on CRI SD 310 is a computer piece by John Melby that makes use of the most widely distributed method of sound synthesis, the MUSIC360 program. The essential configuration of the system is shown in figure 1. *91 PLUS 5* for Brass Quintet and Computer was executed at Princeton University; the digital-to-analog conversion was done at Bell Labs. This piece and another of his works, FORANDRER (appearing on the American Society of University Composers recording) show the judicious use of the medium as regards instrument design and musical presentation. His music is easy to listen to and not without

interest. The clear polyphonic distinctions and timbral differentiations are a welcome relief to the overly complex surfaces in much computer music. The computer instrument (computer macro-instructions assembled to simulate the composer's idea of some instrument) behaves like an actual instrument, so it is logical to use it in conjunction with live instrumental players as in the Brass Quintet piece and what is rumored to be in a future release involving voice and computer.

Since the MUSIC 360 program, as well as its predecessors, are all set up to operate in terms of "instruments" and "notes", its use as a performer of instrumentally conceived music has occurred to several composers. In the case of Melby's piece, the result is the natural outcome of trying to produce a synthesis of live instrumental sounds and computer generated sounds. It has been suggested that the computer might be used to perform music that cannot receive an adequate performance by live performers, or to provide a model for live players, or to satisfy the composer's curiosity about the composition's performance when he has not yet found a viable live performance. Then too, the work may be too difficult for live musicians to handle, given the level of performance of some (or any) players. Benjamin Boretz' *Group Variations* (CRI SD 300) is a piece that originally had a live performance. Owing to the great complexity of the work, and to provide what might be termed an "ideal" or absolutely accurate performance, a computer version was made. The piece is termed "polyphonic ensemble music" and is a relatively difficult piece to get into. The wealth of sonic differentiation and association, only possible with the almost complete control over timbre that the computer offers, makes this piece an interesting one to study, not in the academic sense of analyzing it, but rather in the more casual sense of attempting to "absorb" the piece via repeated approaches. "Now I will listen for this idea and all of its appearances in the 'landscape' of the piece"; "now I will endeavor to relate all the passages that contain such and such an image," etc. The record jacket gives a somewhat provocative, if enigmatic ground-plan for such an attack. The sophisticated use of timbre in the piece must not mislead the listener into thinking of the piece as only a conglomeration of textures, although it can be approached (not too fruitfully) in this manner also.

A very agreeable first experience at listening to computer music might be to examine the *Extensions* for Trumpet and Tape by Charles Dodge (also appearing on CRI SD 300). This piece is also a work for live performer and computer. The nature of the piece is that of a soliloquy for trumpet interrupted by passages of a distinctly contrasting nature for the computer. These passages are all multiple glissandi containing myriads of notes. The remarkable thing about the computer passages is that they consist entirely of sine tones (the simplest waveform possible). If one listens closely he/she can detect the phase interference patterns caused by differing rates of glissando for the various tones. The effect is comparable to the op-art use of Moire' patterns. The ability to control precisely the rates of change of all parameters of a desired note can result in very surprising effects from simple means.

Charles Dodge is perhaps one of the best examples of a composer that is fully committed to technological-artistic interaction. His activities are almost always the result of the quite playful interpretation of some new and as yet unexplored technological vista. In his work *Earth's Magnetic Field* (NONESUCH H-71250) the tri-hourly fluctions of the earth's magnetic field (as measured from several points on the globe and charted on a discrete scale of 28 values) are interpreted as musical pitches. Taking the period of time from January 1st to February 4th of the year 1961 these readings serve as material for a heady composition in C major (side 1). The attractiveness of the piece comes about from a sort of heterophony produced by differing decay rates and slow changes of timbre on various notes of the sequence. The resulting diatonic field produces a strong hypnotic effect that is curiously missing from side 2 where similar treatments, compositionally and timbrally, are used with the scale of values being interpreted as a chromatic (12-tone) scale. Dodge's real claim to fame rests, mainly, not in his work with MUSIC360 (and other similar programs) that he has used at Princeton and Columbia Universities, but rather in his work with speech synthesis.

Speech analysis and synthesis has a history which would be beyond the scope of this article even to summarize. Suffice it to say that fixed formants are the key to the recognition of a timbre as voice-like; i.e., a vocal sound is recognized not by some particular relative relationship of the components of its frequency spectrum, but rather by certain fixed bandwidths that are emphasized no matter what fundamental frequencies occur in the vocal range. Vowel sounds are so characterized. Consonants are produced by transients (sudden changes in the frequency spectrum). An analysis program developed from digital filter technology by Dodge and Richard Garland at Columbia and Kenneth Stieglitz at Princeton partitions a sequence of speech sounds into .01 second summaries or transforms that represent the frequency characteristics for that segment of time. These "windows" are then used to re-synthesize the speech sounds by a process of filtering either a buzz waveform (vocalized speech) or white noise (aspirated speech) with the information in the sequence of "windows" thus restoring the original speech pattern. The windows differentiate between the transient characteristics (consonants or articulation) and the overall timbre and pitch of the voice (vowels). Since this differentiation has been made one can modify the synthesis procedure in the computer by requesting either modifications of the basic vocal quality (say to produce a hi-fi recording of Caruso) to complete substitution of the vocal quality (having Beethoven's Fifth speak Shakespeare). Such a procedure (depicted roughly in fig. 5) allows the composer to construct compositions that have the precise control of MUSIC360-type pieces, but with the added overlay of speech patterns. Thus the image of the human

voice is introduced into computer music. CRI SD 348 contains three pieces of Dodge that utilize the above technique. *The Story of Our Lives* is the most involved work containing a female and male voice (as well as a neutral "book" voice) and has been termed by Dodge an "operatic dialogue." It has been performed as an opera (with actors miming the computer produced speech) and is available on video-tape. The other works on the recording, *In Celebration* and *Speech Songs* are more experimental, but quite humorous speech mosaics.

The technique of filtering complex waveforms to produce new sounds is a built-in part of all sound-synthesis programs and appears as a factor in most of the pieces examined here. An early use of this technique appears on NONESUCH H-71245 in a piece called *Changes* by Dodge. Also included on this recording are pieces by J.K. Randall and Barry Vercoe, the designer of MUSIC360.

The widespread use of MUSIC360 would seem to demand that compositions by its major proponent be available on recordings. So far his *Synthesism* is the only one commercially available. It makes use of all the features of MUSIC360 that motivated its creation. Earlier sound-synthesis programs of a similar nature were less flexible in terms of the user (often a composer with not much computer know-how). The macro-language facility of MUSIC360 and its variants have made it very popular with users. *Synthesism* is a bravura, virtuoso piece demonstrating the capabilities of this improved system. Compositionally the work employs another distinct feature of computer sound-synthesis programs, the ability to generate musical materials algorhythmically.

Vercoe's work has not stopped with MUSIC360. One of his reasons for writing that program was to allow for more rapid and efficient generation of the digitalized sound output. This has been further extended by his work at M.I.T. into a system similar to MUSIC360 that can work in real-time. The system is depicted in highly simplified form in figure 2. The important thing to stress about this new development is that it is very composer-oriented. The score of a work-in-progress is stored in the computer and can be displayed upon demand in terms of a musical score, rather than in the usual numbers representing parameters of musical notes. This allows the user to modify his work without having to use the computer's language — and he can hear the result of any work he does immediately.

Also appearing on NONESUCH H-71245 are three pieces by J.K. Randall of Princeton University each of which makes use of a different version of the original MUSICIV program created at Bell Labs. These programs are slightly more cumbersome to use but essentially operate like MUSIC360. Their main disadvantage is that they consume vast amounts of computer time.

Even with these early versions of MUSICIV the subtly and composerly control of timbre, a hallmark of Randall's computer music, are already strongly in evidence. This is also the experience obtainable from another major work of Randall's, *Lyric Variations* for Violin and Computer (CARDINAL VCS-10057) where timbral relationships between the violin and the computer-produced sounds are exploited. Owing to the relatively large variety of timbral effects possible on the violin and the basic complexity of string tone, the challenge here seems to have involved an attempt to extend the timbral qualities of the violin even further. (One recalls the *Synchronisms* of Davidovsky for live instruments and tape).

Randall's most recent endeavor in computer music composition is his music for the film "Eakins" (CRI SD 328). (Eakins was a Philadelphia artist, a painter of nudes, whose

life and works are examined in the film.) Here we have music that is admittedly "background" in nature. Randall states that it is literally background (in the music-theoretic sense) in that it is not to be noticed even as music in relation to the "film-world" which it inhabits. Something akin to a Paul Weissian kind of definition of music comes to mind: music can be thought of as a kind of abstract matrix that more concrete human events easily fill, in whose ambiguous depths they find a kind of existential time flow that subliminally highlights the surface (foreground) of the "reel" world (in this case) as it passes before us. The effect of hearing this music without any foreground elements (i.e., the film) is very curious indeed. It is mildly sensuous, *extremely* heady music that has a very engaging surface. In concert (several years ago in Town Hall) the piece was even moving despite (or because of?) its very repetitive nature. Randall's computer music is of high artistic merit and he has used the computer with full knowledge of its implications as a medium. (Please read the jacket notes and his *Compose Yourself*, serialized in *Perspectives of New Music*, to gain insight into this remarkable personality.)

Computer technology has had its influence on the electronic music scene in more modest ways. Memory devices that store sequences of control voltages are found on several electronic music synthesizers. These devices, however, provide only a more flexible approach to the organization of musical events, but do not alter the steady-state sounds of analog devices that the ear immediately detects as mechanical and accoustically unconvincing. The ability to modify the timbre of a sound during the course of a single musical event is the real breakthrough in computerized electronics. After all, accoustical instruments as well as the sounds of our environment ("natural" sounds) have a great deal of complexity. With computerized control, electronic music can now start to delve into this relatively unexplored region of music.

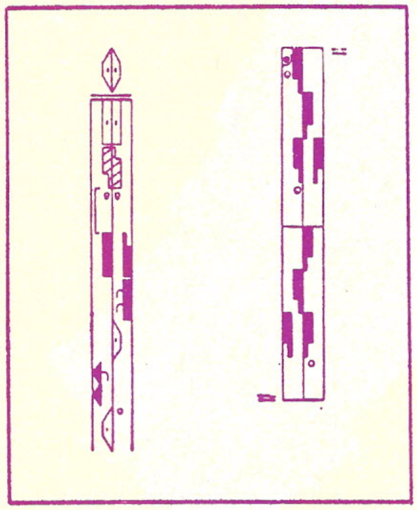
For those who are interested in the use of computerized synthesizer music, FOLKWAYS FTS 33435 offers the efforts of J.D. Robb. Except as a curiosity, this recording is definitely "pour les enfants."

A much more "with it" attempt is to be found on FOLKWAYS FTS 33437 by Jon Appleton. It is a sort of musical travelog (titled "World Music Theatre") evocative of the "hot" medium of the radio (such "tribal" musings no doubt appealing to FOLKWAY'S ethnomusicological bias). This recording should not even be mentioned in a review of computer music except for the fact that a forthcoming release, *The Dartmouth Digital Synthesizer* (FOLKWAYS FTS 33442), which will contain pieces by Appleton and others, was done on the newly completed system at Dartmouth that contains computer controlled digital modules. This system should provide the ultimate in real-time construction of electronic music. (See figure 4). In any case it should once and for all rid us of the flat, two-dimensional sounds of most conventional synthesizer music.

It is difficult to speculate on the future of computer music since its very definition changes daily. With the current rapid increase in technology, possibilities that seemed remote are now happening. Computerized equipment of all descriptions is mushrooming into existence. The most promising developments in technology, both the hardware and programming, are in the area of digital filters and miniaturization. The performance end of electronic music is absorbing developments initiated by researchers and composers with access to larger installations. It is not inconceivable that soon one will be able to purchase a kit to build his own equipment for sound synthesis. The proliferation of electronic instruments with digitally controlled timbres is to be expected. And continuing research both technically and artistically will expand the vocabulary of what we now think of as music to include elements heretofore taken for granted. Texture, timbre, and rhythm especially fall into this category. ■

Dance Notation

By EARL UBELL



Picture Leonard Bernstein rehearsing Beethoven's "Fifth Symphony." There are no music stands; no music scores. Instead, Bernstein teaches each instrumentalist his or her part by picking up and playing each instrument or by humming passages. Slowly he works through the strings, the brass and the tympanists.

Feasible? Yes. Crazy? Of course. No symphony orchestra learns a new piece that way. It would consume vast amounts of rehearsal time. Instead, as we all know, each player sight reads the score. The conductor then molds and polishes the ensemble playing into a work of art.

Yet every ballet and dance company in the world normally stages performances by the first, the crazy method. Choreographers or balletmasters teach each dancer his or her role by the monkey-see-monkey-do technique. Only after hours of stepping and counting can ensemble playing begin. Which is one reason why dancers make about a third the pay of symphony players.

In the past few months, I have witnessed two events that can alter forever the crazy artistic and economic course of the dance.

Event No. 1: A small professional ballet company in Syracuse, N. Y., has learned in three weeks to read written dance scores so that they can now perform any written ballet. This is the beginning of true dance literacy.

Event No. 2: A computer specialist, who is also a composer, has programmed a computer to speed up by at least five-fold the writing down of a ballet. This may be the Gutenberg leap for dance.

In short, these two developments have simplified the reading and writing of choreography to give dancers the equivalent of sheet music.

To those unfamiliar with rehearsal halls, it must come as a shock that one can even write down an ephemeral dance movement or that dance scores exist from which dancers can re-create a stage work. Dance, in its infinite variety,

Earl Ubell, producer of special broadcasts for NBC News, is chairman of the Dance Notation Bureau.

Steps Into A New Era

may seem beyond capture. Arms, hands, fingers, legs, feet, toes, torsos and heads cut through space at variable speeds in any direction in a range of styles from the baroque to the clinical.

One may ask: Why not use a film or videotape to reconstruct a dance? Some companies do so for simple pieces; moving images do give a wonderful overview. But reconstructing from film is like sitting a symphony orchestra in front of a recording and letting the musicians learn the music.

Accurate dance writing has been possible since 1928, when Rudolph Laban, a German engineer, first published the system that now bears his name: Labanotation. In the intervening years, a handful of world specialists have ground out with great effort some 100 ballet scores.

Since Laban, others have created writing systems, notably the late Rudolph Benesh, who devised a method called choreology, but more popularly is called by his name. Benesh Notation dominates British ballet, and an additional 100 works have been recorded by Benesh and his followers all over the world.

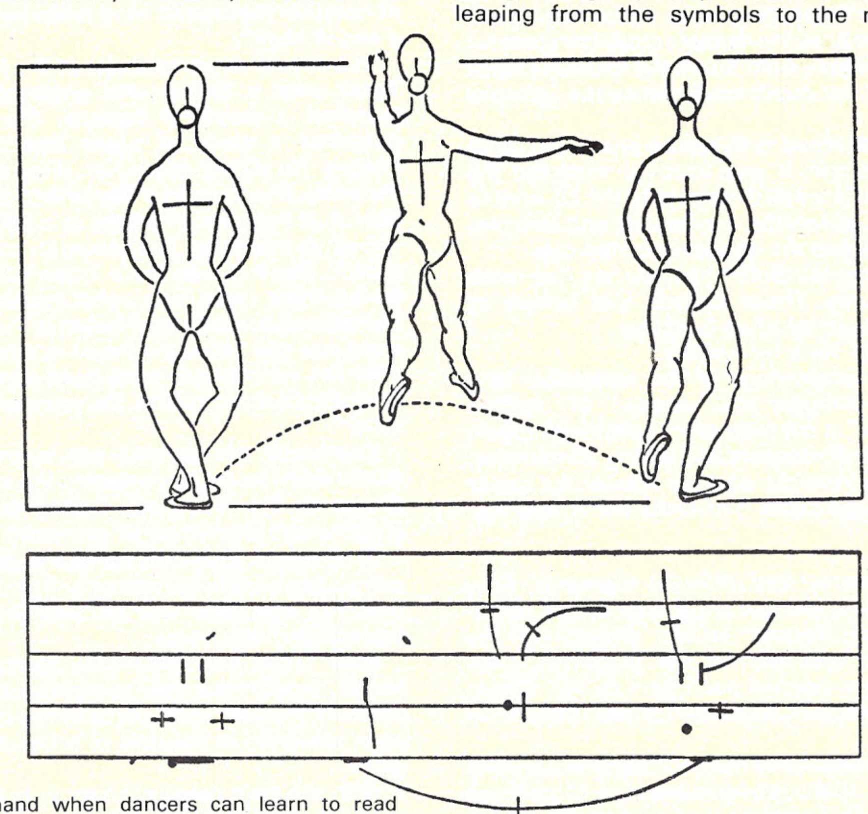
There has been controversy over which system is better. Benesh notators say their techniques are faster and easier to read than Labanotation. The adherents of Labanotation say theirs are more accurate, more detailed and more responsive to the choreographer's intentions. The truth? Opinions are easy to come by, but there is no valid test that has established the ascendancy of one system over the other.

Just last season, New York saw four Kurt Jooss Ballets, 40 years old, revived with the aid of Labanotation scores. (The choreographer's daughter, who remembered her father's works, also played an important role in the production.) Ten Labanotated works by Doris Humphrey, who died in 1958, play each year to larger audiences than when she was alive. "Shakers," her pulsating dance piece about the religious sect, throbs on stages from Kodiak, Alaska, to Australia, to New York.

Despite the triumphs of notation in keeping some dance alive, only a few dancers can read a Labanotation or Benesh score, and no prominent choreographer writes one.

For many dancers nothing can replace the actual movement; not even film or videotape. They believe dance exists only in the dancer's body at the moment of performance. For them writting seems to freeze the artistry.

But time and technology can change all that. For example, about three years ago, Muriel Topaz, the chief Laban notator of the Dance Notation Bureau, a nonprofit organization devoted to spreading the word about notation, happened to be in France taking the French course for foreigners at the Sorbonne. "We did not translate the French into English," she recalls. "I learned the French words for objects by observing them — 'hands,' 'forks,' 'books' — saying 'les mains,' 'les forchettes,' 'les livres.' In a month I was dreaming in French." It struck Miss Topaz that this teaching method could be transferred to reading notation. Accordingly, when she returned to New York, she plunged a group of professional dancers into reading, leaping from the symbols to the movement — rarely



The time may be at hand when dancers can learn to read choreography recorded in Labanotation (top) or Benesh Notation with no more difficulty than learning a foreign language.

From "An Introduction to Benesh Dance Notation," published by Adam & Charles Black; "Principles of Dance and Movement Notation," published by Dance Horizons

stopping to explain the meaning of the symbols in words. Since the dancers were professionals, they easily performed the movements. Soon they *saw* the symbols as movements.

The test came when Tony and Sirpa Salatino, directors of the Syracuse Ballet Theater, invited Miss Topaz to use her new teaching method to teach their seven-member company to read notation and translate it into dance. The Salatinos felt that if their company could learn to read, they could have easy and inexpensive access to an enormous ballet repertory that exists as notated scores.

So Miss Topaz flew to Syracuse for a week where she and Allan Miles, another notator, taught the dancers five to six hours a day to read and perform. At the end of a week, she handed the company the score for "Pas de Trois" by Andre Eglevsky (after George Balanchine) with music by Michael Glinka. The idea was to have the dancers study the score for the next week. Miss Topaz would return the following week to see whether or not her pupils could dance the Eglevsky work.

When Miss Topaz returned to Syracuse she found three casts had learned the trio. It was in rough form but the seven dancers had mastered all the moves of the 15-minute piece. She had only to polish the movements and interpret the few notation passages the dancers puzzled over.

Since then, the Syracuse Ballet has learned five more works in the same way. These include four ballets by Antony Tudor and a work by Anna Sokolow.

So much for reading. Now for writing.

To understand the importance of the computer, let's follow Miss Topaz through the notation of Jerome Robbins's "Les Noces," a complex story ballet with music by Stravinsky involving 26 dancers and lasting 25 minutes. She worked with Lucy Venable, another notator, for a year to produce the 365-page score that looks like a long-lost Aztec codex.

The two notators transcribed different segments. They attended 75 hours of rehearsals conducted by the ballet-master James Moore, former artistic director of the Swedish Ballet, who had learned the piece from Mr. Robbins. Both notators learned each part as Mr. Moore taught it and made their own shorthand notes, unreadable by anybody else.

Then each notator transcribed her shorthand into careful pencil sketches. After all the rehearsals ended, Miss Topaz took several months to complete and correct the pencil score. After several more months, the corrected score came back to the Dance Notation Bureau for an orthographer to make a carefully inked version. The inking took about 200 hours. (Today the bureau has a typewriter with a ball-like element created by I.B.M. for typing scores twice as fast as inking.)

Altogether more than 1,000 hours went into the project at a cost of more than $4,000.

Dr. Stephen Smoliar, a professor of computer science at the Moore School of Engineering at the University of Pennsylvania, who happens to be a composer and dance aficionado, and Maxine Brown, a graduate student there, have now made possible even greater use of the computer.

Today Miss Topaz can sit before a television screen that is connected to a large computer. She touches a type-writer-like keyboard and rotates some knobs, and the notation glyphs appear on the screen, perfectly drawn and in proper position. In a few minutes she can fill the screen with a page of notation that would have taken her hours to complete as a pencil sketch.

In a recent demonstration at the Ford Foundation, Miss Topaz used a Tektrotronix graphics terminal. When she punched a couple of buttons, the terminal printed in moments a piece of paper that contained the screen images. No inking! Moreover, if the operator wishes to edit the score, she touches a few keys to eliminate a symbol or to alter it. It happens in tenths of a second.

Even at this stage, the computer could eliminate hours of tedium. But it can do more. It can do a lot of rote checking. For example, there is a symbol that designates a bent torso. Sometimes a notator will forget to "unbend" the torso. The computer can be programmed to recognize the symbol and after the notator has completed a page to print out: "You have a bent torso symbol. Do you wish to unbend or will you wait?"

Eventually, the terminal can be moved to the rehearsal studio with the notator entering the symbols directly on the screen at the rehearsal. During rehearsal, the choreographer can receive paper printouts of work.

Dr. Smoliar says that eventually the computer program can accommodate Benesh or any other notation. Indeed, in time the computer will be able to translate Laban into Benesh and vice versa.

Miss Topaz estimates that notating a ballet like 'Les Noces" would have been completed in less than 200 hours and a nearly complete, partially checked score would have been available at the end of rehearsals instead of a year later.

With the twin developments in reading and writing, we are raising the dance to the status of the other performing arts in giving it a continuous documented history, and making it cheaper to rehearse, perform and establish repertory. ■

NEW HORIZONS FOR MICROCOMPUTER MUSIC

Malcolm Wright[1]

Since October of 1974 when the first 8-bit micro processor kit was introduced to the hobbyist, the computer kit market has exploded with a variety of supporting peripheral circuits. Who would have guessed that today a person could have his own personal computer at home to generate a form of animation on the television screen, play games in a software language like BASIC, control home appliances like a burglar alarm, or produce different frequencies to an audio amplifier in the form of music? All this, and yet few of the applications or potentials of the micro computer kit have been developed.

One of the applications for the micro computer which is just starting to be explored by the hobbyist is music. With only 45 bytes of instruction a simple routine was written by Paul Mork which could read a table of binary numbers in memory and generate a square wave frequency related to the value of the numbers. The small program by Paul could play simple melodies like "Daisy" or "Jingle Bells" when executed. Due to the speed of the micro computer, frequencies up to 2000 cycles per second could be produced.

By December 1975 the scope of software music was expanded again. Alpha-numeric music with amplitude control was introduced by *PCC*[2] in a magazine article. The author, Malcolm Wright had written Alpha-numeric music for the 8080 with capabilities not considered before. The music was still a coded table of bytes for each melody, but the bytes were alpha-numeric characters in the ASCII format. Now to play a note like middle C, the user just typed "4C" which specified the octave and the note. If the user wanted a sixteenth note of B flat, one octave higher, he would type "5SB!" into memory. Alpha-numeric music allowed the user to vary the volume, tempo, duration of the notes, generation of rests, repeat measures in music, generate six octaves of notes — sharp or flat, and create envelopes (attack) for special tonal qualities.

Software has its limits and many companies are in the prototype stage of developing computer control hardware devices for music. The modern electronic organ with a band box (rhythm generator) gives an idea of how far we can go in synthesis of instrument sound. Imagine an instrument like the MOOG synthesizer, used in many electronic music recordings, controlled by a computer! Dr. Prentis Knowlton in Pasadena, California has interfaced a PDP8 minicomputer with a pipe organ and with the assistance of many interested friends has encoded musical pieces like Bach's *Concerto in A-Minor* and Rimsky-Korsakov's *Flight of the Bumble Bee* for computer control. The Bumble Bee can be played at tremendous speeds by the computer with no mistakes and with complete repeatability.[3]

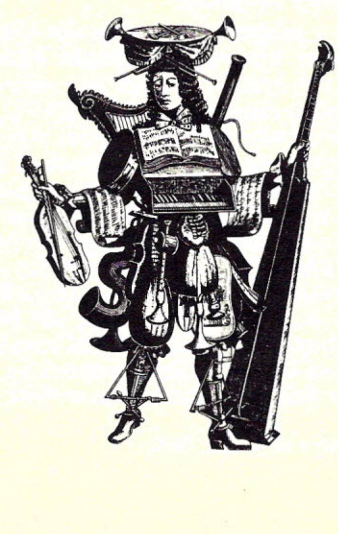
If one is going to generate electronic sounds from a special circuit board for a computer, what should be some of its capabilities? The circuit should be able to simulate different tonal qualities by generating different complex waveforms other than just sine or square waves. The circuit board should be able to give different attack and decay times for the notes to realistically simulate the various kinds of musical instruments. The frequency range of the circuit should be the whole audio spectrum from 15 cps to 20,000 cps at the minimum. The user should have control of the volume and the duration of the notes generated.

Another requirement that should be placed on the music synthesizer circuit board is that a minimum of the computer's time should be used to control the card — less than 50%. If the control time is less than 100% computer usage then the computer can be executing other programs at the same time. Imagine computer games with sound effects in the background at the same time!

The future for computer controlled instruments or synthesizer sounds is exciting. There are at least three companies presently developing these kinds of products. Solid State Music in Santa Clara, California is presently prototyping an Altair compatible card which will meet all of the above requirements.

As a last note, imagine the future composer being able to write and edit pieces of music for a whole orchestra and being able to play the music instantly after completion by typing RUN on his computer! ■

[1]Solid State Music, 2102A Walsh Ave., Santa Clara, CA 95050
[2]Peoples Computer Company, Box 310, Menlo Park, CA 94025
[3]An LP record of this system, "Unplayed by Human Hands" is available for $6.98 from Computer Humanities, 2310 El Moreno St., LaCrescenta, CA 91214.



MPL: A PROGRAM LIBRARY FOR MUSICAL DATA PROCESSING

by Gary Nelson*

MPL (Musical Program Library) is a comprehensive system for processing musical information in a digital computer. Since music exists as graphic symbols and abstract concepts as well as acoustical phenomena, MPL provides facilities for working in each of these domains and in their intersections. MPL includes programs for sound synthesis, computer assisted analysis and composition of music, computer assisted instruction, and automated music printing.

Hardware

The core of the Oberlin College Computer Music System is a Xerox (now Honeywell) SIGMA 9 computer. The SIGMA 9 operates in time sharing, batch, and real time modes. The particular advantages and capabilities of each mode are exploited in various operations of the MPL system. Communications between the SIGMA 9 and the Computer Music Studio (CMS) are carried out via a Datamedia 1520A APL/ASCII video terminal and via the Oberlin College time sharing network which is connected to terminals of various types at sites all over the campus. Sound is produced through a real time systems interface which contains four 15-bit digital-to-analog converters and the related analog circuitry required to transform the computer generated signals into high fidelity music. CMS is also equipped with professional quality audio tape decks and loudspeakers for recording and monitoring the output of the conversion system. Musical graphics are produced on a Calcomp 563 incremental plotter, a Tektronix 4013 graphics terminal, and a Diablo Hyterm II impact printer/plotter. A diagram of the hardware configuration of CMS may be seen in Figure 1.

Host Programming Language

The choice of a programming language upon which to base MPL became the most important decision in the development of the system. Preliminary versions of MPL (under other names) were attempted in BASIC, FORTRAN, SNOBOL4, and PL/I with varying degrees of success. Each of these languages was rejected because of inefficiency, incompleteness of implementation, limited availability, or incompatability with processes which were considered basic to a sophisticated musical programming system. Although FORTRAN has been retained for those components which require optimum computing efficiency, the majority of MPL is now written in APL (A Programming Language). APL has been described by its inventor Kenneth Iverson (1962) and in an excellent textbook by Gilman and Rose (1976). To borrow an informal remark from Andy (James A.) Moorer of the Center for Computer Research in Music and Acoustics at Stanford University, "better programming languages make better music languages." APL is certainly a better programming language than those mentioned earlier and its adoption has immeasurably advanced the development of MPL. APL is a dynamic language which is at home in the interactive environment of modern time sharing computer systems and is easy to learn in spite of its initial cryptic appearance. APL is also growing in popularity and may soon become the major language in computing. In short, many of the features which are described below as essential attributes of a musical programming system are basic capabilities of APL.

Sound Synthesis Program

The sound synthesis program for MPL is called FMORCH (an acronym for frequency modulation orchestra). FMORCH

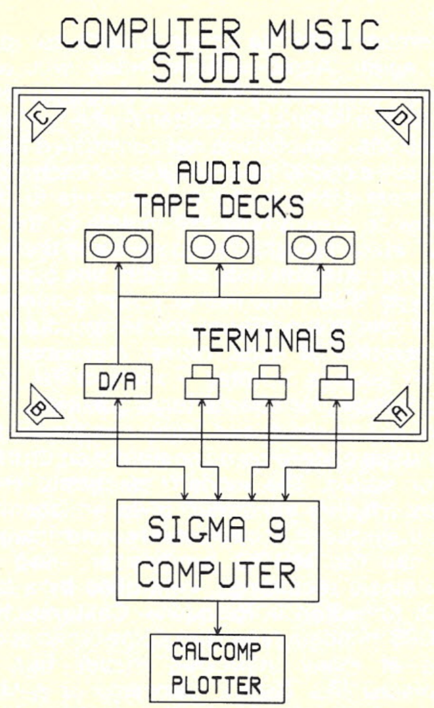


Figure 1. Hardware configuration of the Oberlin Computer Music System.

*Oberlin Conservatory of Music, Ohio.

computes the time varying pressure functions of the signals which represent specific musical sounds. This technique is well known and has been described in detail by Mathews (1969). Because this method of digital sound synthesis involves the computation of at least 40,000 numbers per second of sound, speed of processsing is critical. For this reason, FMORCH is written in FORTRAN and METASYMBOL (the Xerox assembly language). The FORTRAN portion of FMORCH is an extension of sound synthesis programs written and described by Howe (1975a, 1975b).

Structure of the Orchestra. FMORCH contains 64 instruments and is capable of performing compositions with 64 simultaneously sounding parts. These parts are distributed as they are needed over as many as 32 musical lines. Each line may contain any number of parts (up to a combined maximum of 64) and that number may change every millisecond if the MPL user so specifies. Each line may be independent of all other lines with respect to dynamics, articulation, location and distance panning, reverberant characteristics, and timbral quality. FMORCH performs in a hypothetical acoustical space in which the exact center is designated 0 (see Figure 2). Four loudspeakers are positioned in a square in which the corners are a distance of one unit from the center of the space. Distance in MPL is expressed on a scale of 0 to 16 where 16 represents a theoretical distance which is 16 times the distance from the center of the space to any one of the speakers. Location is expressed in degrees of a circle whose center is point 0. Location 0 is midway between speakers A and B. Figure 2 illustrates the path of a sound which pans from 0 to 720 degrees (two clockwise revolutions) while moving from distance 1 to 3.

Synthesis techniques. FMORCH uses two techniques described by Chowning (1971, 1973). The frequency modulation (FM) technique is the most elegant, efficient, and versatile method yet developed for digital sound synthesis. FM is capable of producing quite acceptable simulations of existing musical instruments along with a wealth of timbral effects which do not exist in nature. All of the instruments in FMORCH are connected to a reverberating mixer. In this mixer, the output of each instrument is processed through a network of amplitude scalers and recursive filters to simulate natural reverberation and cues for location and distance. The implementation of these two techniques in FMORCH includes facilities for dealing with loudness contours, harmonic spectra, formant characteristics, and noise transients. Although an extensive understanding of acoustics is required to deal with the full capabilities of FMORCH, the methods provided for describing timbral parameters are designed with the musician in mind.

Portable FMORCH. An equivalent version of FMORCH is being prepared entirely in FORTRAN so that it may be exported along with the remainder of the MPL system. This version will be compatible with MUSIC4BF (Howe, 1975a) and will require only minor modifications in that program.

Notation Program

The MPL program for musical notation is a generalized package for processing graphic data. This program reads symbol codes which are produced by the MPL graphics editor and generates the appropriate sequence of pen movements for each character to be drawn by the Calcomp plotter. For proof reading, the output of this program may be directed to the Tektronix 4013 graphics terminal. Although the screen on this terminal is too small to provide a satisfactory representation of an entire page of musical score, it is sufficient for detecting misplaced characters and similar errors. The Tektronix terminal is also useful for viewing fragments of a score page when detailed editing is required. An alternate version of this program which will produce output suitable for the Diablo Hyterm II impact plotter/printer is under development. The MPL graphics program is capable of generating the standard set of musical characters (see Figure 3) and special symbols which may be designed and encoded by the MPL user. All of the figures for this paper were produced with MPL graphics programs.

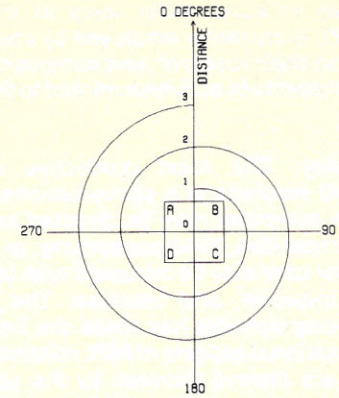


Figure 2. Trajectory of a sound traveling from distance 1, location 0, to distance 3, location 720 (two revolutions).

Design Considerations

The goals for MPL were stringent and perhaps even idealistic at the outset. These goals revolved around the fundamental precept that musical considerations should never be sacrificed because of hardware limitations or the expediencies that tend to arise from the application of the artificial languages of computers to human activities and thought processes. The initial concepts of MPL have remained unchanged through seven years of research, four preliminary versions, and five host programming languages. Nearly all of the original concepts have materialized in the present version of MPL. The following is a summary and brief discussion of those attributes which were deemed essential to the development of a sophisticated tool for computer applications in music.

Integration of Applications. As mentioned above, music exists in more than one domain. Computer programs for music which favor sound synthesis over analytical capabilities or musical graphics over compositional operations will quickly discourage the imaginative musician. He will be led through a maze of incompatibilities among diverse programs and systems and inevitably reach the common but unfounded conclusion that computers are basically unresponsive to musicians. MPL strives to remove this confusion by confronting the user with a single entity which is capable of treating the well established musical applications of computers in a comprehensive and integrated manner. This characteristic of MPL helps to promote the highly desirable state in which the relationships and interactions between the various musical domains may be better understood and exploited.

Musical Style. MPL is intended for use by music theorists, historians, performers, music educators, and ethnomusicologists as well as by composers who expressed themselves in a wide range of musical styles. It is paramount therefore that MPL be without prejudice with respect to musical style. Initial use of MPL has been encouraging on this issue. The earliest applications of MPL include demonstrations of Javanese and Persian scales, performances of Baroque keyboard works in Werkmeister intonation, and the production of rhythmic and harmonic

exercises for use in aural skills tests at the Oberlin Conservatory. MPL is currently employed by about a dozen faculty members in their research and composition and by more than thirty students in activities related to their course work.

Conversationality. The most productive interaction between man and machine is a conversational mode. In such a mode, the computer may be directed to carry out tasks which are repetitive, time consuming, or utilitarian while the computer user may be occupied with tasks which require insight, initiative, and intuition. The advent of modern time sharing systems has made the implementation of the conversational aspects of MPL relatively routine. In MPL, actions are always initiated by the user. When responding to such actions, MPL will frequently lead the user by requesting information that is required to carry out the prescribed operation. MPL may also report intermediate results and ask for a decision on how to proceed after some part of an operation has been completed. Although the program will occasionally appear to be in control, the user has several means by which to escape that control and resume the initiative. It is a conversation in which the person must always have the last word.

Transportability. In the belief that computer music has seen too much activity that may be compared to the reinvention of the wheel and the rediscovery of fire, MPL has been designed to be exported to other sites and implemented on a wide variety of computers. The choice of APL as the host language contributes greatly to the transportability of MPL. Initial investigation indicates that only minor alterations in the Oberlin implementation of MPL will be needed to install the system on IBM, UNIVAC, Burroughs, DEC, and Control Data computers. The author is currently working with members of the National Consortium for Computer Based Musical Instruction to install and verify MPL on each of the most widely used computing systems.

Maintainability. Any complex system for data processing must be easily maintained and updated. MPL is designed in the best tradition of structured programming. It is entirely modular and each module performs a single well defined function. Adding new features to MPL and exterminating bugs in old features has proved exceptionally easy in the early stages of the use of the system. APL provides debugging facilities which are interactive and extremely powerful when compared to similar facilities in other computer languages. The file management components of MPL are straightforward and the structure of MPL data files is simple and direct. Each entry in a file is affixed with a code which indicates what kind of data is contained therein and how that data relates temporally to other data in the file.

Learnability. Since MPL is intended for musicians, it is designed to be intelligible first with respect to the terminology and experiences which are familiar to individuals with musical training. All components and operations in MPL have musical identifications. The interactive nature of APL further contributes to speedy comprehension of the capabilities and use of MPL. A facility is provided with which a user may ask for a description or tutorial example of a particular feature of MPL. APL provides a facility with which the program designer may detect errors made by the user and initiate some corrective action. If such an action cannot be taken, the designer may replace the standard APL error messages with more informative messages of his own. Such messages in MPL often contain suggestions for tutorial exercises which will lead the user toward an understanding of the nature and cause of the error. In short courses with musically sophisticated users, six hours of lecture/demonstrations proved to be sufficient introduction to MPL so individual work could be undertaken. Conservatory students are exposed to MPL in a semester course which includes many tutorial exercises and a complete introduction to APL.

Expansibility. The initial experiences of the MPL user can be accomplished without a knowledge of APL. Although it is not recommended, an individual may accomplish a great deal musically with MPL without learning APL. The use to date has shown that the versatility of MPL increases directly in proportion to the user's familiarity with APL. In the computer music course at Oberlin, APL and MPL are taught concurrently. In addition to this inherent expansibility of the system, MPL contains features which encourage communication among users. A directory of MPL users is maintained with annotations describing the nature of the work each user is doing. Suggestion and mailbox functions are also provided. Functions are available to enable users to share data and operations which they have designed for their own application of MPL. It is hoped that this ease of communications among users of diverse musical interests will increase the quality and quantity of work in computer music.

System Organization

A chart of the relationships among the principal components of MPL may be seen in Figure 4. The following is a brief description of each of these components and how they interact in MPL.

Workspace. The APL workspace is a hybrid entity which contains both programs and data. In MPL, the workspace acts as a control center and as a scratch pad. The control operations are concerned with communication between the workspace and the MPL data files. The workspace contains functions with which the user may create and

Figure 3. Example of musical score printed using the MPL NOTATE function.

manage his personal files. A variety of input and output functions are available and parameter codes are provided so that the contents of data files may be addressed by type and temporal location rather than by absolute indices. When operating as a scratch pad, the MPL workspace may be used to experiment with musical structures. Once these structures are developed to the satisfaction of the user they may be transferred to a data file. Musical data structures may be manipulated analytically and compositionally in a variety of ways by using the large vocabulary of functions provided with MPL. The MPL user may combine these functions into larger and more specific operations with which to carry out his musical computations. Since initializing an MPL workspace is a very complex procedure that a typical user cannot be expected to deal with directly, an initialization program is provided to get the user started and to give him introductory information.

Score File. The MPL score file contains data which describes a specific composition. This file is organized by lines which correspond to the lines described in the discussion of FMORCH. The score file has 99 lines which are individually addressable by setting a line pointer. (Only the first 32 lines may be performed by FMORCH.) Once the line pointer is set to a particular line, all data transferred from the workspace to the score file is associated with that line. The line pointer may be changed as often as the user desires and data may be entered into the score file in any order. Within a line in the score file, the storage location is determined by temporal information in the data. The unit of measure for time in MPL is the beat. A beat is an arbitrary unit which has been given the value 1. Each line may contain up to 9999 beats. Each entry in the score file corresponds to a beat and all musical events which occur at or near a particular beat are stored together in the file. This scheme results in a chronological ordering of the score file which eliminates time consuming sort operations and facilitates the work of the score editor. The data stored in a score file relates to pitch, rhythm, tempo, dynamics, articulation, and instrumentation. For compositions which are to be performed by FMORCH, the score file also contains specifications for location and distance panning and for characteristics of reverberation such as depth and decay time. For compositions which are to be printed, the score file contains specifications for clefs, key signatures, barlines, time signatures, and other graphic musical data. The MPL user may create a new score file for each composition or he may store several compositions in the same file through manipulation of the line pointer.

Timbre File. MPL timbre files contain information which describes the acoustical characteristics of musical instruments. Each parameter may develop dynamically during a note or musical phrase and that development may be dependent on or independent of other timbral parameters at the discretion of the MPL user. Each entry in the timbre file represents the prototype of a musical instrument. Multiple copies of an instrument are automatically generated whenever two or more simultaneously sounding notes are assigned to the same timbre. The definition of a timbre therefore makes possible an ensemble of up to 64 instrumental parts with that timbre. Each MPL user may have his own timbre file and there is a provision for users to share timbres by mutual agreement. In addition, a public timbre library is available to all MPL users.

Graphic File. MPL graphic files contain information describing the pen movements required to draw a character or symbol on one of the graphic output devices. Each MPL user may have his own graphic files and/or obtain graphic symbols for his musical score from a public graphic file. The public graphic file contains all of the standard musical

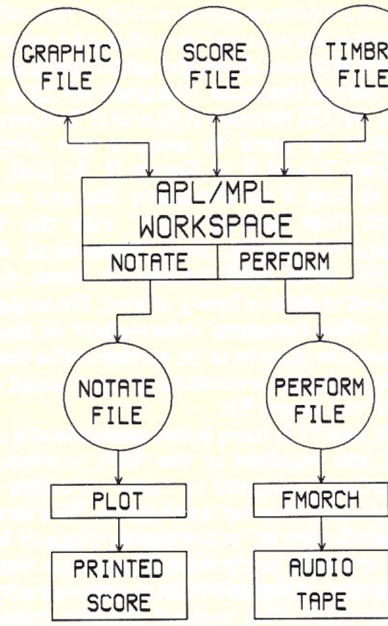


Figure 4. Organizational chart of the files and programs in the MPL system.

characters such as duration symbols (half notes, etc.), clefs, accidentals, barlines, and dynamic markings.

Score Editor. The MPL score editor is made up of functions which permit the selective insertion, deletion, or replacement of data in a score file. Figure 5 shows the sequence of MPL operations required to encode the musical excerpt shown in Figure 3.

Discussion of Figure 5. The following remarks are in reference to the numbered lines of Figure 5 and are intended to illustrate the use of MPL through an elementary example of musical encoding. At (1) and (2), H3 (three half notes) and DQE3 (three repetitions of a dotted quarter and eighth note pattern) are defined for later use. H3 and DQE3 are duration lists in which the beat has been interpreted as a quarter note and durations are expressed relative to that interpretation. If negative durations had been used, they would be interpreted as rests by any MPL function which operates on duration lists. The beat in MPL is an arbitrary unit of time which is subject to interpretation by the user within the context of a particular composition. A composition in 3/8, for instance, would be encoded more easily if the beat were interpreted as an eighth note or dotted quarter. Note the use of REP in (2). REP is one of a number of MPL functions which provide shortened methods of encoding musical data. At (3), the function MEL is invoked to define the "melody" L1. A melody in MPL is simply a musical structure in which each duration has only one pitch (as opposed to chords and does not have any esthetic connotations. MEL asks for a list of durations in (4) and issues the prompt character in (5). In (6), the user supplies a duration list using the predefined fragment H3 along with numbers which specify the nonrepetive durations of line 1 of Figure 3. At (7), MEL has counted the number of durations entered and informs the user that 9 pitches are required and issues a prompt at (8). At (9), the user responds with a pitch list expressed in MPL pitch codes. In the MPL

pitch code, 48 is middle C (C4) and a tempered semitone (100 cents) is 1. An octave is 12 semitones; therefore, the first pitch in (9) is an octave above middle C (60=48+12). If a pitch code contains a fractional part, that part is understood to be the number of cents above the pitch designated in the whole number part of the code. The pitch 48.23 is 23 cents above middle C. In (10) through (16) and (17) through (23) a similar procedure is used to encode the pitches and durations of lines 2 and 3 of Figure 3. At (24), the line pointer is set to line 1 of the score file and data to be associated with that line is "put" into the file. The parameter codes PIT (pitch), DYN (dynamics), KEY (key signature), CLEF, and MET (meter or time signature) indicate what kind of data is being stored. These parameter codes combine with temporal information in the data to determine where the data is to be stored in the file. At (27-32) and (33-38) a similar procedure is used to put data into lines 2 and 3 of the score file.

After (3), (10), and (17) have been executed the variables L1, L2 and L3 are resident in the MPL workspace. The structure of the data assigned to these variables is a four column matrix with a row for each pitch. This structure is designated a "pitch matrix" in the terminology of MPL. The first column of an MPL pitch matrix contains the starting times of the notes as computed from durational information supplied by the user. The second column contains the durations of each note. The third and fourth columns contain the initial and terminal pitch for the note. These two pitch columns are identical except in the instance of a glissando. MPL contains a wide variety of functions for operating on pitch matrices.

Timbre Editor. The timbre editor provides the facility for dealing selectively with the individual parameters of musical timbre. The development of each parameter over the duration of note is specified as a line segment function. The number of line segments may be different for each parameter. The duration of each segment may be independent of (real time) or dependent on (relative time) the duration of the note to which it is being applied.

Since timbre definition is a rather complex matter which requires a reasonable acoustical insight, MPL functions are provided to lead the user through tutorial sequences in instrument design. MPL timbres may be given names which are descriptive of the sound they produce. The turn around time on the Oberlin computer system is also short enough to make trial and error a viable approach to the specification of timbres.

A complete description of the timbral capabilities of MPL is beyond the scope of the present discussion. However, a few brief highlights will give some insight into the interrelationships among the parameters of timbre. Duration may be a function of pitch in such instruments as the piano (with the sustain pedal depressed) or cymbals where resonance will decrease naturally over time according to the physical characteristics of the vibrating body. Larger bodies tend to vibrate longer than smaller bodies of the same material and construction. It has been made possible therefore to define a timbre in MPL which lengthens or shortens the durations of notes according to some scale of simulated resonances. This permits synthesis of such effects as *laissez vibre* which is common in compositions for harp and chimes. Pitch is a function of location and distance when a sound is traveling fast enough to produce the Doppler effect. The harmonic spectrum of many traditional musical instruments varies according to the loudness and pitch of the note being played. Vibrato rate often increases in a crescendo. A listener's perception of harmonic spectrum is influenced by his distance from the source of the sound. Such parameters as pitch inflection and noise transients are common in acoustical instruments and desirable in synthetic instruments if the intent is to simulate the natural qualities of live performance.

Graphic Editor. The MPL graphic editor permits the user the ability to deal with the visible manifestions of music on several levels. On the lowest level, he may specify the pen movements of the plotter directly. A character may also be defined interactively at the terminal by designing a dot matrix. Once such a matrix is completed, a function may be invoked to translate the dot pattern into a sequence of pen movements. Characters may be concatenated to produce composite symbols. Symbols may be moved to any X-Y coordinate and their size may be altered independently in the X and Y dimensions to produce elongated and elevated characters. Symbols may also be rotated through a full 360 degrees.

PERFORM Function. The MPL function PERFORM merges a score file and a timbre file into a performance file which is suitable as input to FMORCH. With PERFORM, the MPL user may proofread a score or timbre aurally. PERFORM permits the performance of selected lines within selected time frames as specified by the user. With this capability, the user may isolate and perform particular passages in his music without having to start at the beginning each time. Short performances of thirty seconds or less can usually be accomplished while the user is in the Computer Music Studio. Longer performances of up to six minutes are run overnight and stored on digital tape until they can be processed through the digital to analog converter. Several of these overnight runs may be spliced together to form extended compositions after the conversion has taken place and the music resides on audio tape. PERFORM also supplies appropriate default values for parameters which the user leaves unspecified in the score. The defaults include location and distance, reverberation depth, and even a default timbre with which the user may perform a score before he develops the skills required to design his own instruments.

NOTATE Function. The MPL NOTATE function merges a score file and a graphic file into a notation file which is suitable as input to PLOT. (PLOT is a small FORTRAN program which is required because no direct communication between APL and the Calcomp plotter is available at

```
       H3 <- 2 2 2                                        (1)
       DQE3 <- 3 REP 1.5 .5                               (2)
       L1 <- MEL                                          (3)
DUR?                                                      (4)
::                                                        (5)
       H3,4 2,H3,4                                        (6)
9 PITCHES?                                                (7)
::                                                        (8)
       60 60 60 61 60 58 58 58 63                         (9)
       L2 <- MEL                                          (10)
DUR?                                                      (11)
::                                                        (12)
       H3,DQE3,H3,4                                       (13)
13 PITCHES?                                               (14)
::                                                        (15)
       (4 REP 56),57 56 55 53 56,4 REP 55                 (16)
       L3 <- MEL                                          (17)
DUR?                                                      (18)
::                                                        (19)
       DQE3,4 2,DQE3,4                                    (20)
15 PITCHES?                                               (21)
::                                                        (22)
       41 42 41 39 37 36 34 46 39 41 39 38 39 37 36       (23)
       LINE 1                                             (24)
       PIT PUT L1                                         (25)
       DYN PUT 0 6 6 7 12 6 18 7                          (26)
       CLEF PUT 0,TREBLE                                  (27)
       KEY PUT 0 -4                                       (28)
       MET PUT 0 3 2                                      (29)
       LINE 2                                             (30)
       PIT PUT L2                                         (31)
       DYN PUT 0 6 6 7 12 6 18 7                          (32)
       KEY PUT 0 -4                                       (33)
       CLEF PUT 0,ALTO                                    (34)
       MET PUT 0 3 2                                      (35)
       LINE 3                                             (36)
       PIT PUT L3                                         (37)
       CLEF PUT 0,BASS                                    (38)
       MET PUT 0 3 2                                      (39)
       KEY PUT 0 -4                                       (40)
       DYN PUT C 7 6 6 12 7 18 6                          (41)
```

Figure 5. Example of an encoding sequence using MPL score editing functions.

present.) NOTATE interprets the numeric information in the score file and translates that information into the equivalent graphic symbols. Once the translation is complete, NOTATE formats the score by positioning each character according to the established rules for musical notation. Figure 3 was printed by applying NOTATE to a score file created (in part) by the execution of the MPL program shown in Figure 5.

Conclusion

It is hoped that the preceding discussion of the philosophy and mechanics of MPL will provide some preliminary understanding and appreciation of the capabilities of the Oberlin Computer Music System. Although MPL is installed and working, its ultimate test is still in progress. That test is of its utility to a wide range of musicians from students to professionals whose musical interests and ideas cover a territory which is perhaps too large to measure. ■

References

Chowning, J.M., "The Simulation of Moving Sound Sources," *Journal of the Audio Engineering Society*, 1971, *19*(1), 2-6.

Chowning, J.M., "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation," *Journal of the Audio Engineering Society*, 1973, *21*(7), 526-534.

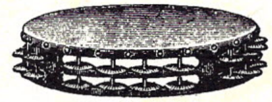
Gilman, L., & Rose, A.J. *APL: An Interactive Approach*. New York: Wiley, 1976.

Howe, H.S. *Electronic Music Synthesis: Concepts, Facilities, Techniques*. New York: W.W. Norton, 1975a.

Howe, H.S. *MUSIC7 Reference Manual*. Flushing, New York: Queens College Press, 1975b.

Iverson, K.E. *A Programming Language*. New York: Wiley, 1962.

Mathews, M.V. *The Technology of Computer Music*. Boston: MIT Press, 1969.



COMP SCI SERENADE

(Sung to the tune of
"My Bonny Lies Over the Ocean")

My program lies under the backlog
My card deck's all over the floor
The plotter is using a crayon
And I just can't take any more

CHORUS:
Bring out, bring out
Oh bring out my printout today, today
Bring out, bring out
The one you ripped off yesterday

The card reader chewed up my job card
And someone erased all my files
The system has been down for hours
While people collapse in the aisles

CHORUS:
Flunk out, flunk out
I worked like a dog each and every day
Flunk out, flunk out
Twelve projects were due yesterday

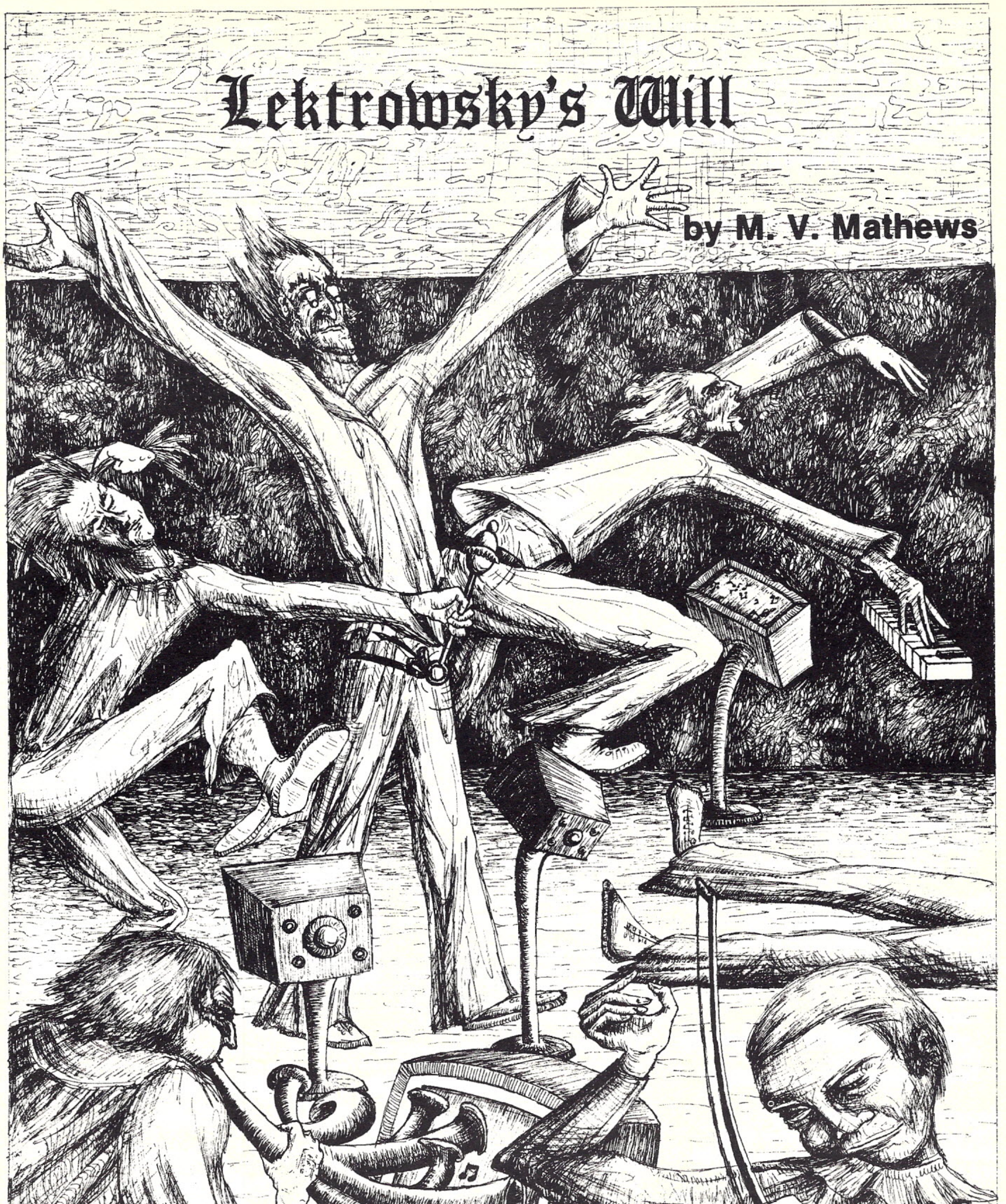
Security holes I've discovered
The records of grades are now mine
What once was a one-point-five average
Will soon be a three-point-nine-nine

CHORUS:
Send out, send out
Oh send out the grades to big companies
Send out, send out
They'll all want a scholar like me!

**— Terry Bollinger & The Watt Five
(Computer Science Dept.
Univ. of Missouri—Rolla)**

Lektrowsky's Will

by M. V. Mathews



"The future may be a fiction."
— Anonymous survivor of a New Jersey mugging

— New Universe Spaces, June 13, 2165 —

John Lektrowsky, the world's only STARNAUT, joined the Chewlard Order yesterday. The massive gates of the students' portal of the Chewlard Monastery opened briefly to receive his body and instrument. He entered clad in the traditional "Tails" with violin, the 1704 Betts Stradivarius under right arm, Tourte bow dangling from thumb and forefinger, and the page of manuscript in left hand. Observers speculated that it was the opening of the Back Chaconne, but, of course, this information is not revealed by the Order.

Lektrowsky's sensational return from Alpha Centauri preempted the headlines nine months ago. He was the first and only STARNAUT to be declared sane on arrival, although some consider his subsequent application to the Chewlards at the certified age of 35 to be an indication of, at least, judgmental disequilibrium. Others attribute it merely to long practice or time alienation from the world. Lektrowsky, himself, maintained that his choice was a reasonable consequence of 20 years acceleration during which space he practiced the Chaconne and that only the impossibility of playing the music preserved his motivation and sanity. Whether or not his opinions are believed, the Chewlards broke their traditional rule limiting novitiates to five years of age or younger. The argument over Lektrowsky's age has never been satisfactorily resolved. He was born August 14, 1985, entered space in the year 2001, and returned in 2164, earth time. According to both his records and the certifying physiologists, he spent twenty life years under acceleration attaining a speed greater than 99 percent of the speed of light. Physicists still have not been able to explain the time discrepancy. One group maintains the theory of relativity must be revised, another that he passed close to a Black Hole.

Lektrowsky left a will to be made public in two weeks.

— New Universe Spaces, June 27, 2165 —

STARNAUT'S WILL READ

John Lektrowsky's will was communicated today from the offices of Burk and Fint in an unusual transmission.

Lektrowsky is assumed to be still alive somewhere in the Chewlard auditoria, though his existence will be con-

firmed only if his style can be identified on a recording which, as is well known, are the only messages ever to leave the monastery gates. Although public interest in Lektrowsky has somewhat abated, Burk and Fint's communications were copied by all major news media. The most unusual feature was the will. The only property involved is the information in the document itself and this "wisdom" Lektrowsky left to the entire universe in order that "the future may profit from the experience of the past". Following the *Spaces* longstanding policy, the entire document will be published in the Moonday Supplement and annotated excerpts are given below.

Will of John Lektrowsky

I, John Lektrowsky, being of sound mind but unusual experience and having voluntarily left the company of man by joining the realms of eternal Chewlard practice, do nevertheless feel kindly toward the world and wish to give to it the wisdom which I, the only sane and surviving STARNAUT, possess. Accordingly, I hereby will this document to the entire Universe for whatever good uses can be made of these unique opinions. Since time is short, I will simply attempt to describe my reactions, those of a 20th century man, to the 22nd century world. My comments focus on music, because both the world and I owe our sanity to these vibrations. Their unique effect on our brains has never been explained or understood and I can subtract nothing from the mystery, but it is clear that without this form of expression race suicide might end our great civilization, even as apparently happened to the strange earless creatures on the inner planet of Alpha Centauri.

The need in every man to create something beautiful, at least to his senses, was not appreciated in my day, perhaps because it was partly fulfilled in the course of normal work. A lucky man might spend his years building houses or boats or gardens, which could be made more beautiful, or at least better, by the loving effort he put into their construction. Even the automobile worker could be proud of the sparkling chromium and bright colors on his cars knowing that somewhere deep inside he had tightened a vital nut to just the right torque, and

Editors note . . .
Max Mathews is considered to be "the *Father* of Computer Music" by virtually everyone now working in the field. Currently, he is the Director of the Acoustical and Behavioral Research Center at Bell Telephone Laboratories. This lab does work in the areas of learning, speech communication, vision, psycholinguistics scaling, sensory physiology and physical acoustics.

At Stamford's Artificial Intelligence Laboratory I was given a grand tour by Leland Smith, Dick Moore, and John Chowning of the important, fascinating work they're doing with computer music. Dick Moore, who worked with Max Mathews at Bell Labs, knew of the existence of *Lektrowsky's Will* and he and I felt it would afford *Creative's* readers a unique opportunity to glimpse a fictional future of music by someone who today is creating that future. Mr. Mathews granted us permission to print this work.

—BG

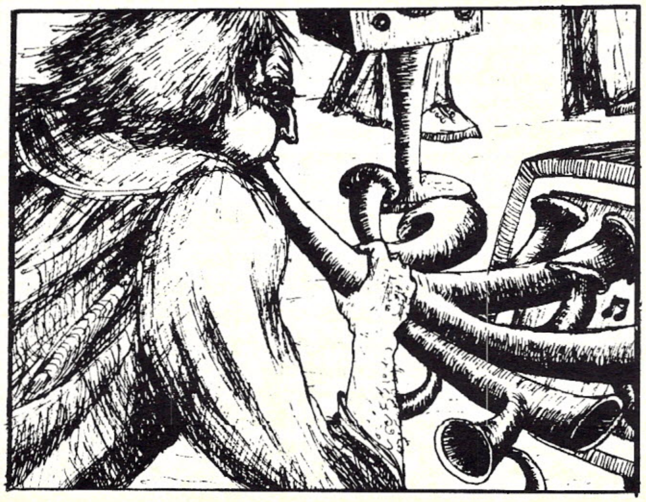
that the motor might fall out if his work was neglected. *(Editor's note — the automobile, an ancient form of transportation, was used until about 2010 when the petroleum supplies were exhausted.)* Now, with automatic factories, polyethylene grass, and the one-month work year, a sense of creativity in work is difficult to feel. To some extent this need has been filled by creativity in the arts and especially music where the new instruments and trainers have made it possible for almost anyone or any group to play a self-satisfying auditory performance. *(Editor's note — Here the will continues at some length developing the argument that, with the outlawing of any form of live recording, auditory performance is pure creativity and can have no utility, hence it has rightfully been excluded from productivity rationing. Music is contrasted to painting where no satisfactory method for disposing of finished pictures has yet been invented, and they continue to pollute the walls of our buildings.)*

On Quadraspeakers

My star ship having been equipped with an excellent stereo system, I was somewhat surprised to learn that recordings were reproduced over only four loudspeakers, albeit excellent ones. I had really expected to find the walls of rooms papered with tens or even hundreds of speakers designed to attack the ear from every conceivable direction. Of course, I had no way of knowing about the quadraspeaker revolution which occurred in the year 1990 and was decisive, if brief.

At that time, manufacturers started phasing out the dual stereo tapes and their accompanying four speakers and making available only incompatible eight channel disks which required completely new turntables plus, of course, four more speakers. The response from the public, whose memories of the transition from two to four speakers were all too vivid, was immediate. Factory walls were stormed, production lines wrecked, and atrocities committed. An unfortunate Vice President of Advertising, who tried to sway the mob, was found with his head in a folded horn driven by a 500 watt amplifier playing both channels of an old stereo Rock recording.

Happily, a technical solution to the impasse was feasible. Some work of an early computer musician, Howning, was rediscovered and developed. By using ingenious techniques involving mixing reverberated and direct sounds plus Doppler frequency shifts plus time delays, Howning was able to demonstrate that four speakers are sufficient to reproduce sounds from any direction or any distance and to create moving sounds, in fact to create sounds moving faster than the speed of sound. These methods were developed into the quadraspeakers which have been standard for so long.



On the Absent Audience and Virtuosity

Some men, returning from a long trip, yearn for ice cream or some other edible. I, coming home from an incredibly longer journey, dreamed of again hearing a concert, even the nonmusical parts of the performance — the orchestra tuning, the audience coughing, the conductor tapping his baton. You can imagine my crushing disappointment to learn that audiences had been banned many years ago from all auditory performances. Only gradually have I come to agree with the wisdom of this decision. I now understand that any live performance of any Index work would be bound to be far inferior to the flawless recordings, released in cathedra by the Ecumusical Synod, and reproduced to noiseless technical perfection over the quadraspeakers built into every room. *(Editor's note — Although no one would argue against the superiority of recordings, the audience ban was enacted for another reason — to protect the performers. Auditory performance grew out of an ancient ritual, The Happening, invented in the mid 20th century by Allen Kupro, John Kage, and several other artists far in advance of their time, perhaps too far. The instruments available then were so poor that the result was frequently unendurable and performers were all too often lynched by the angry audience.)*

Even after hearing a recording, I cherished the hope that occasionally a monastery gate might open a crack to allow a select few to witness a recording session. My naivete was so great I had to be told that any virtuoso worthy of his tails would play at least twice as fast as the listening tempo and all recordings are carefully slowed down before being released. Paginenius, according to my informer — a man with rank of Mabbot — played nothing slower than four times real time including his measure from "The Flight of the Bumblebee". Such virtuosity, he explained, is achieved by having each performer concentrate on a short section, typically a measure, of one piece and practice it to the ultimate perfection. Complete performances are created by abutting the efforts of many such specialists. While I look forward to such an opportunity to perfect my technique, I must admit to occasional nostalgic memories of my voyage, when I played the entire Chaconne, however badly. *(Editor's note — Certain restrictions are presently in effect to limit ways of achieving virtue. Paginenius resulted from the crossbreeding of a Congolese drummer with a Chinese ping-pong champion. Such genetic engineering is outlawed.)*

On Audio Performance

To you, Audio Performance — Audance as it is called — must have the comfortable familiarity of a close friend, but for me, coming from a time when each note had to be individually handcrafted, my first Audance seemed a miracle of mass production in which swarms of notes grouped and regrouped themselves to embody the performers' musical ideas effortlessly as if in answer to their very thoughts. As I later learned, the computer was controlled by perfectly ordinary devices and the "miracle" lay in the trainer's program. I am most grateful to the government for making an exception to the strict laws banning audiences at Audances, though I feel it was entirely justified by my special circumstances. I believe my disguise as a repairman looking for an intermittent bug in the computer was accepted and the performance was normal.

I arrived a little before the performers. The chamber, a pleasant room almost 15 meters long was furnished with about a dozen consoles resembling TV sets distributed in a comfortably irregular pattern. Some had chairs, others were at standing height so the performer could move freely as he reacted to the music. Only the control console of the computer and the card reader were actually in the

room, the main circuits being in an adjacent room. The usual quadraspeakers were incorporated into the walls and the sound came from there just as it would have for any recording.

From my chosen post, hovering over the card reader with miniscope in hand, I was able to watch the players assemble. All but one selected some control devices from a cabinet and plugged cables from these into their consoles. A box with knobs, an organ keyboard, a wand which could be freely moved in three dimensions, or a set of foot pedals were popular with the younger players. Some devices looked vaguely like archaic instruments — a violin without a body, a clarinet with a solid tube, a board which could be struck with a stick. These were favored by the older players who I guessed might have studied traditional instruments.

In discussing the performance afterwards with the trainer, I learned that the simplicity of appearance of the devices was deceptive. The knobs responded not only to twisting, but also to how hard they were grasped and even to the skin temperature of the performer. The keyboard was sensitive to both vertical and horizontal pressure on the keys as well as to velocity and displacement of key stroke. As far as the computer was concerned almost any device could control any function so that the players could quite arbitrarily select something that fitted their training or mood.

The player who had rejected all mechanical devices merely stood in front of his console and watched the TV display intently, thus making me erroneously suspect him of being some sort of performance critic. *(Editor's note — Criticism of audio performance is one of the few capital crimes in our society. Fortunately, it is rare, the last critic having been executed over 50 years ago.)* Far from being a critic, the trainer explained that this player, one of the most advanced in the group, was a looker. Built into each console is an eye-tracking camera, hence it is quite possible to play any note displayed on the TV screen simply by looking at it. With a more complex program, groups of notes can be played by sweeping one's eyes across them. Although eyeing a score is one of the most facile performance techniques, more eye training is required to become proficient than might be imagined and not many performers reach this advanced state.

An immediate quiet in the group was produced by the entrance of the trainer, a beautiful girl with long yellow hair, whose presence made it difficult to concentrate on the card reader. Trainers, in my time, would have been called composers, or conductor-composers, except that they would not have had the superb skill in programming achieved by the trainer through long years of understudying master programmers nor the required certificate in psychotherapy.

The Audance began with some discussion, not entirely audible to me, amongst the performers and trainer in which, I believe, she ascertained the mood of the group. Next she selected a deck from her large music case and dropped it into the card reader. At the same time the performers took up their various devices. The trainer next inserted a key in the computer console and a low sound gradually swept round the quadraspeakers, rising in intensity, pitch, and rhythmic modulation as it moved. The effect on the performers was immediate and utter. During the set-up time, I had to attend discretely to my miniscope and poke the card reader occasionally to assuage some slightly worried glances. From the first vibration of the Audance to its reluctant finale, I could have paced in Chewlardian Tails in front of the oblivious performers, so completely was their attention captured by the sounds they were creating. The trainer explained that one of her most critical functions was to resolve and

terminate the performance while the players were still alive. Early Audances, before trainers were mandatory, sometimes got into man-machine loops which were broken only by the collapse of the player, or in fortunate cases by some computer error.

Three and one half hours later, as the last sound reluctantly died into a reverberated distance far beyond the walls of the room, the players slowly laid down their instruments and slipped from the room guiding their feet over familiar steps with unfocused eyes. They had returned from another world, an inner world immeasurably further away than Alpha Centauri.

How did the music affect me? Technically, there was no question about its excellence, and indeed there could hardly be any since the compositional rules were part of the program and the computer would allow exceptions only on command of the trainer. The sound quality was also superb. All the normal instruments could be heard at various times when the brilliance we have come to expect from their enhanced reproductions on the quadraspeakers. But in addition, many new sounds were incorporated — some so different from normal instruments that it would be useless to attempt to describe them, others which seemed like crosses between standard instruments. I amused myself inventing names for some — the Trumpolin, the Obow, the Piananet, the Harpsibone. My fantasies were closer to the truth than I realized. Players could construct timbres by mixing traditional instruments, for example, 10% violin, 30% trombone, and 60% tympani. The classic instruments provided a convenient palette to be combined into these new sounds. The three-dimensional wand was a popular tone control.

However, comments on technique and sound quality are procrastination to delay answering my original question which must now be faced. Without risking violation of the criticism laws, I can certainly say I was not moved by the performance as deeply as were the performers. Many times the music rose to great peaks and fell beautifully into the intervening valleys which are as necessary as the peaks themselves to define the summits. But the succession of climaxes did not form, for me, a convincing landscape. I do not know the reason — perhaps the environment was constructed by the trainer for other personalities, perhaps the players' improvisations limited the organization, perhaps my móod was unreceptive. In any case, I did not regret the non-recording act which, as the Audance ended, condemned the last never-to-be-repeated vibration to oblivion. But the thoughtful departure of the players testified to memories which would be long

cherished. *(Editor's note — The Spaces take no responsibility for the above statements which border dangerously close to criticism. Fortunately, the arm of the law seldom reaches inside the monastery walls.)*

On Trainers

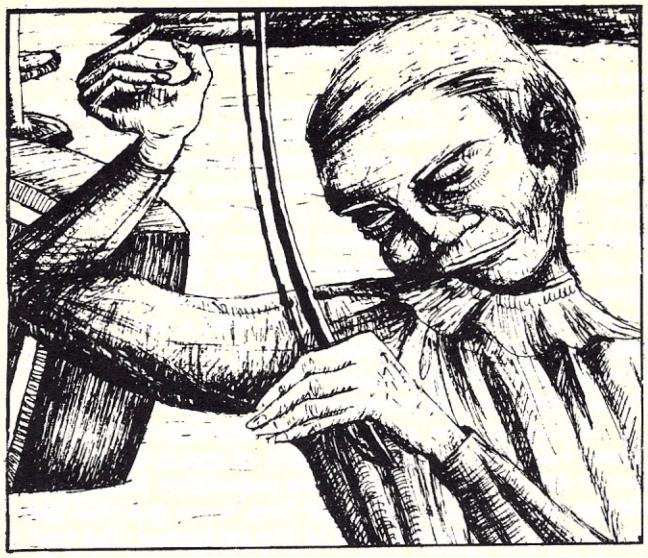
The skills of a trainer which encompass both an explicit theory of music and the most sophisticated programming techniques could hardly be described here even if I understood them. I can only recount a few recollections of an after-the-performance drink with Alison who immediately denied that she composed "works of art". In fact, she put down those pompous old soundbags like Beethoven and Brahms whose delusions of grandeur let them write something so it could be played again and again. Her function was to create an environment in which her group could express themselves musically in ways satisfactory to themselves. This, she maintained, was both far more difficult than making a composition to be played by expert musicians and much more valuable than endlessly repeating the same old notes. Of course, she admitted the old fogies had perhaps done as well as they could considering their crude instruments.

By contrast, the computer allowed almost any possibility from completely unrestricted improvisation to the exact replication of precomposed notes, though she would only consider precomposition for very inexperienced performers. Frequently, she would supply the harmonies by means of a program to a melody improvised by one of the players. The harmony rules could be changed from time to time, either by herself or one of the players. Alternatively, the computer might improvise the melody, and the players, as the spirit moved them, might improvise restrictions on it making it proceed rapidly or slowly, in scales or in great melodic leaps, or in arpeggios.

The most interesting, but trickiest, controls involved interactions among several players. One might create the durations of a note sequence while another made the pitches, or one might create a rhythm pattern and the second a counter rhythm to go with it. When necessary, the computer would resolve any conflicts according to its rules by changing the notes of one or the other player. Even so, Alison had to carefully select compatible partners and, not infrequently, change the rules or arrange a hasty divorce to avoid a breakdown in the music.

The TV consoles were invaluable in controlling the Audance. They could show a selection of possible sounds which the performer could play in any order he wished or, with another subroutine, they could show a phrase precomposed by Alison which the performer could start or stop at any time. They could reveal to one performer what another was improvising or show one of the global functions which Alison often provided to control the climaxes and valleys of the grand structure. In rare cases, they could picture a staff with notes that were played in the manner of ancient performance, except that the players could relax knowing that the computer would correct any mistakes they might make. I have already mentioned the unique use of the TV by the lookers. Alison confirmed my feeling that without TV, Audance would scarcely be possible.

In my century a woman composer was a rarity so I inquired whether Alison felt out of place in a male profession. Her reply was delayed by a sharp spasm which she was able to suppress with some difficulty by holding her napkin over her mouth. When she had recovered her voice, she explained that although the Men's Equality League occasionally cried discrimination or picketed an Audance, very few men were able to compete as trainers. She didn't know exactly why — something in the feminine outlook — men just never seemed to be able to learn



how to properly maintain the climaxes so vital to musical finales.

We parted with her suggestion that, although group Audances were relaxing recreation, she also led private performances, which were incomparably more satisfying, and if I would come by her studio some evening, she was sure she could make me forget the violin. Unfortunately, I had an appointment with the Chewlardian High Mabbot on the following day and, by the time I again thought of Alison, my future was committed in other directions. Had it not been for this quirk of timing, I suspect things would have come out very differently and I might not now be writing this will.

On Monastic Music

As far as I have been able to determine, no layman has ever before interviewed a High Mabbott in any of the musical orders. Not only was I granted this privilege, but the Mabbot answered all my questions and appeared to withhold nothing. It almost seemed as if he wished to communicate a better understanding of the Order to the world through me. Why he chose this poor vehicle I cannot say except that the customary audition at the beginning of the interview went well. As the demonstration I had chosen the quintuple stops at the beginning of the second measure of the Back Chaconne, having practiced these from my second through thirteenth years in space. As written, all tones must be struck with precise simultaneity. This, I achieved during final acceleration to the speed of light, and though the nonrelativistic performance at earth speeds is immensely more difficult, nevertheless the Mabbot was well pleased. *(Editor's note — W. A. Back, who is occasionally confused with J. S. Bach, lived in the remote Green Forest of Bavaria in the 17th century. His music was lost and rediscovered in 1981 in the mens room of the New York Public Library on some old paper pressed into service during the great paper shortage which resulted from the Sierra Club's legislation forbidding all tree cutting. In the Green Forest, during Back's time, a five string varient of the violin was popular, which greatly simplifies performing some of his music.)*

The Chewlards contain three ranks of virtuosi, Performers, who are rightfully the only group entitled to wear tails, Archivists, and Untouchables. The duties of the Performers, as one would expect, are to practice their notes diligently and, when requested, perform flawlessly for the recording microphones.

The Archivists maintain the Index of Classics and the

master digital recordings of all classics. In order to be put on the immortal Index, a composition must be unanimously canonized by the Ecumusical Synod of Archivists. Thereafter, a digital recording of the definitive performance will be kept on quartz plates in the archives, the Untouchables will be authorized to issue records and all performances will be counted. The Chewlards, unlike some orders, do not maintain that the universe will end when all classics have been perfectly played 2440 times. They do, however, keep a careful count of all performances. Needless to say, few compositions achieve the Index, in fact, none have been canonized in the last century for reasons that are controversial. Some maintain that audio performance has diverted the interest of composers away from classics, others feel that the reason lies with the unplayable (and unlistenable) music composed at the end of the 20th century by the successors of the 12-tone school. In any case, since no limit is ever put on playing speed, the existing classics are sufficient to provide an eternal challenge to the Performers.

Even if the art of performance would vanish, the classics would be perpetually preserved on the quartz archives located deep inside a granite cave cut into the heart of a mountain whose location was unknown even to the High Mabbot. On these plates are written in sputtered gold, numerical samples of the four sacred signals used to drive the quadraspeakers. Each signal is sampled 440,000 times per second, and each sample is quantized into a 440 digit binary number, so to human senses the recordings are flawless. But, in addition, the samples contain not only check sums, but also error correcting digits, so that errors are unthinkable. In only one case was the perfection of the archives challenged. In the year 2051, the 126,532,543th sample of Beethoven's Ninth Symphony changed from 124613 . . . to 124615 . . . due to a most unlikely constellation of flyspecks. A High Synod was called immediately to deal with the emergency, but fortunately before they acted, the difficulty was cured by an ingenious Archivist who washed the plate.

The role of the Untouchables at first seemed enigmatic within the purity of the musical Orders. Their function is to handle the unclean electronic apparatus used to record performance, slow the speed to the proper listening tempo, abut and mix the various performers and compute the quadrasignals. This last operation included such enhancements as reverberation, filtering and noise stripping whose existence is seldom mentioned to the performers. But, as the Mabbot so clearly put it, one must eat to play and the income of the Order comes from record sales. Hence, it must compete in quality with the other sounds coming from the quadraspeakers. I was moved by the beauty and directness of his wisdom.

A Farewell Warning Against Misinterpretation

The day of my entry into the Chewlards dawns and I must end this will with one caveat. My choice should in no way be interpreted as evidence that classical performed and recorded music is superior to audio performance. A man is a prisoner of his age and, in fact, is bound to what he has learned as a child. In my time, before Telespeak became universal, different peoples spoke many languages. Although it was possible for an adult to learn a new language, he could never master it in the way he would have learned it as a child. Today's music is a new language for me, and though I can intellectually appreciate its power, yet in another sense I can never learn it. Audance is the privilege of the youth of today and I must seek the music of my youth, which happily is also a rich language. ■

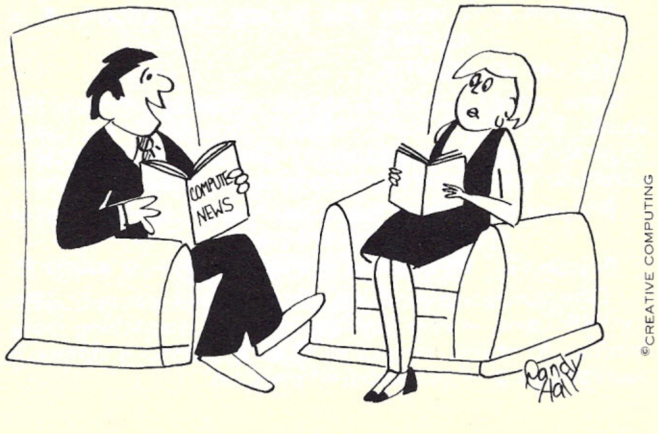
Three New Microprocessors and Floppy Disk Controller from Intel

For the last 12 to 18 months Intel Corp. has watched its overwhelming lead in the microprocessor market get slowly chipped away by such products as the Zilog Z-80 and Motorola 6800 as well as by second source suppliers for the 8080 itself. In the low end of the market competitors like Fairchild, Mostek, National Semi, and TI have been making real inroads.

Intel, however, has been spending $10 million per year on microprocessor R&D alone, more than the total annual sales of many of its competitors. The results of this massive research program are beginning to reach the market and more are on the way. Here now:

- *8085 MPU.* Object code is fully compatable with the 8080A and bus-compatable with 8080 components, the 8085 operates with standard speed memory at 3mHz instead of the 8080's 2mHz. The 8085 operates on a single + 5 v supply for all components. A complete system can be built around 3 chips instead of 10 or more required by 8080 systems. The 8085 appears to be squarely aimed at the Z-80 and 6800.
- *8048 MPU.* A simplified and cheaper version of the 8080. Can operate as a single chip MPU since both ROM and RAM memory are on the chip itself. Aimed at appliance control and office equipment markets. Main competitors are the Fairchild F-8 and other low-end stuff.
- *8748 MPU.* Another single chip MPU but equipped with erasable programmable ROM (PROM) instead of just ROM. This is certainly the most innovative of the three MPUs; main applications will be in control environments although with the PROM it will be suitable for low-volume or tailored products.
- *Floppy Disc Controller.* This is the first of 17 (yes, seventeen!) other LSI circuits that Intel will introduce in 1977. Functions range from peripheral controllers to various interfaces which replace a board full of chips.

Even with these new products Intel will have a real battle on its hands to maintain or gain position. National Semi just down the road now markets an 8080 equivalent with a full set of support circuits. But no matter which of the manufacturers comes out on top it looks like it's going to be an exciting year for the end user. ■



"Wow! The centerfold this month is the new IBM Triplex 304!"

MUSIC ANALYSIS: SLAM SIMPLIFIED OR HOW THE COMPUTER COMPARES 16TH CENTURY BOURGEOIS WITH EIGHTEENTH CENTURY BACH

by
Thomas G. Whitney*

Step back a few years to The Ohio State University into a situation now not at all unique to that institution.

1. We had a body of computer readable musical compositions.

2. We had several computer operations for the analysis of music.

3. Upon looking beyond our own institution, we found several encoding schemes in vogue for situation one. Similarly there was overlap between institutions in the operations which had been computerized, especially for generating scores, counting notes, and classifying chords. Some of these operations, however, were "limited" or more intent upon a particular composer, time period, or geographical region.

What was needed was some easy way to adapt computer-analytic operations to a variety of compositions. For this purpose I developed the computer command language called SLAM (*S*imple *L*anguage for *A*nalyzing *M*usic).

To show what kinds of analyses have been done and are being done, I present the following examples. All words in the solutions which are printed completely in upper case are words "known" to SLAM.

In that which follows, any explicit reference to numbers 1 through 371 refer to J. S. Bach's chorales. The SLAM User's Manual (available upon request) gives an index of the composition numbers in *SLAM's* library.

Problem A: I would like to see how chorale 6 is encoded.

Solution A: PRINT 6.

If one wishes to look at a specific subset of a chorale, the words SOPRANO, ALTO, TENOR, and BASS are used. Several encoding systems permit reference within SLAM to particular instruments or even sections of an orchestra.

Problem B: I would like to see how the soprano part of chorale 9 is encoded.

Solution B: RETAIN the SOPRANO part in 9; PRINT IT.

In the above two solutions, one could substitute SCORE for PRINT and receive from the computer something more familiar to many musicians. This is merely a different representation of the same information.

*The Ohio State University, Columbus, OH.

Problem C: I want to see the score for chorale 6 after it is transposed to D major

Solution C: TRANSPOSE 6 to D major; SCORE IT.

So far the examples have been minimally analytic. Many are familiar with the concept of computers used for counting.

Problem D: I want to see what kind of chords Bach used in chorale 6.

Solution D: COUNT CHORDS in 6.

As output this sentence will list for each vertical simultaneity found a) the pitches by part, b) the duration, c) the chord type: triad (major, minor, diminished, or augmented) or seventh chords (7th major major, 7th major minor, etc.), d) the root if "chordal," e) the Roman numeral, f) the figured bass notation. Information is also tabulated in terms of total duration of each type of simultaneity (as well as number of instances), each root, and root progressions.

Other harmonic measures include TENSION and DENSITY.

No example yet given has taken more than one half of one second of computer time to complete.

Suppose someone at another university was interested in the chordal analysis operation mentioned above. Suppose further that the musical composition was not Bach's chorale but a piece by Chopin and the encoding language is MUSTRAN—2.

Problem E: I want a chordal analysis of a composition numbered 888 which is by Chopin and encoded in MUSTRAN - 2.

Solution E: TRANSLATE 888 from MUSTRAN - 2 to MUSICODE - A; COUNT CHORDS in IT.

Here one may see the possibility that any composition in another encoding language can be converted (translated) to an encoding language expected by a particular operation in *SLAM*. Hence, one can augment to *SLAM* compositions in other encoding systems provided that computerized translators exist for this. These translators are possible because most encoding systems have ways to consistently represent the four essential local properties of a note: the letter name, the accidental, the octave register, and the duration.

SLAM can also accommodate operations which have been developed for other encoding systems. Hence, *SLAM*

is open-ended in the operations, compositions, and encoding systems available to it.

With this introductory overview, we can now return directly to some musical examples.

Problem F: I would like to know how many and what kinds of melodic intervals there are in the soprano part of chorale 235.

Solution F: RETAIN the SOPRANO part in 235; COUNT all FIRST order melodic intervals in IT.

This output would include the quality, size, and direction of all intervals and all possible combinations of those properties of intervals. Furthermore, one can also elect to have the source (starting note) of the interval given. This melodic interval operation has been shown to be quite useful in discrimination studies. In particular the higher order (defined in the next paragraph) melodic intervals with respect to time and country tend to be adequate discriminators when "properly" combined.

An Nth order interval can be defined as having N minus one consecutive notes. By default, a zeroth order interval counting operation merely tabulates the attacks and durations of notes. Similarly a second order interval operation considers all existing sequences of three consecutive notes for the composition(s) in question.

Problem G: Compare the melodic lines of a few compositions from the 16th century with a few from the 18th century.

Solution G: RETAIN in 218, 229, 284, 209, 237, and 244 the SOPRANO part; COUNT all melodic INTERVALS in IT; TABULATE IT; 218, 229, 284 = 165 / 209, 237, 244 = 185

In the above one is given a hint of the ability to recode or renumber sets of data. The tabulation will give two columns of data "165" and "185" for the sixteenth-century and eighteenth-century data respectively. It so happens that the two sets of compositions can be used for contrasting authors as well since the first three are by Bourgeois and the last three are by Bach. To be sure, this would not constitute a definitive study, unless we increase the size of the sample.

In addition, *SLAM* users have a number of compositions and operations for twelve-tone music. Some of these operations will become apparent in the next three examples.

Problem H: Find all twelve-tone rows in Webern's Opus 28

Solution H: FIND in 491 all LINEAR rows.

Composition number 491 is Webern's Opus 28, Movement 1, in *SLAM's* current library. This operation will find all twelve-tone rows or aggregates. If any row found is a permutation of the original row (supplied in the bibliographic information at the time of encoding), inversion of the original, retrograde inversion, or simply the retrograde, this will be so indicated. If it is none of these (i.e., a simple aggregate of twelve notes), this will also be indicated. One may ask for notes fewer than twelve in number. For example one could ask for all ten-note aggregates in this composition.

Problem I: I would like to find all twelve-tone rows in any combination (vertically or linearly) in that same composition.

Solution I: FIND in 491 all VERT-LINEAR rows.

Here again one could have asked for fewer than twelve notes if desired. A possible solution (output) from this operation could be the indication of four consecutive vertical simultaneities of three notes each. Again all rows found are tested for possible permutations of the original row, or its inversion, and so on.

One can look for embedding of rows within rows. This can be done in both the linear mode and the vertical-linear mode.

Problem J: I want to find all rows in that same composition. Upon finding a row, I do not want that row as a candidate for subsequent rows.

Solution J: FIND in 491 all LINEAR rows GIVING 801. REMOVE from 801 all ROWS found GIVING 802. FIND in 802 all LINEAR rows.

One can repeat these operations as often as desired.

Maximum overlap is permitted so that all existing rows (or aggregates) are found.

Several of the analytic problems of music are not unique to music. Consider the next two examples which are also computerized operations in literary data processing.

Problem K: Compare the different settings by Bach of a melody by Hans Leo Hassler.

Solution K: RETAIN in 21, 74, 80, 89, 98, and 270 the SOPRANO part; TRANSPOSE IT to C major; EXPAND IT; COLLATE IT.

The chorales are in different keys so the transposition has been introduced to avoid letter name variation from setting to setting. The EXPAND has been introduced to allow for the repeat signs in a few of these chorales. This eliminates the repeat signs and produces a long string of music or notes as heard not as scored. The COLLATE is the actual borrowing from the literary people. This is an operation they use to compare variations in different editions of the same text or manuscript. In music the concept of word insertion or word deletion is bound by time. It takes the form of a quarter note becoming two eighths or vice versa. In music the total duration of the composition is generally preserved.

Problem L: I want to study the use of all notes in the soprano line in context for chorales 1, 2, 8, and 9.

Solution L: RETAIN in 1, 2, 8, and 9 the SOPRANO part; CONCORD IT.

Concording is done by certain literary scholars to study author traits in word usage in context. It is frequently done by dictionary makers as well. The CONCORD operation gives a center-of-the-page listing of notes (alphabetized) with notes appearing to the left being notes which come before it in time and similarly the notes which follow the given note in time come to the right. An identification of the composition is also given. The operation also lists the absolute statistical frequency of each note and gives a most frequent to least frequent listing as well.

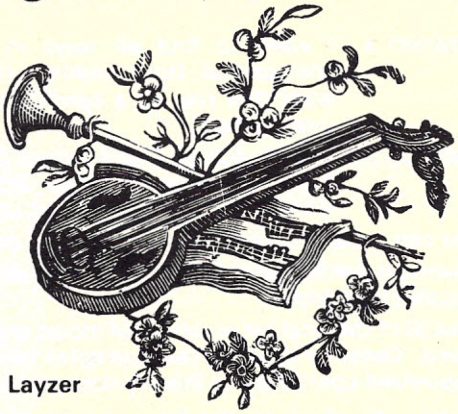
The output of the chordal analysis operation (and density or tension operations as well) can be input data to the concording operation whereby one finds the use of vertical structures in context.

There are several subsetting operations within *SLAM*. The user may have several hundred compositions under investigation in any command in *SLAM*. From these one could select those with a particular key signature, time signature, or mode. In addition, one can take subsets of particular bars of a composition. In a given command one could take even 30 or 40 or more subsets of the input data. There are many more operations available in *SLAM* which I will not go into now.

A few years ago I developed an interactive form of *SLAM* called *MAP* (*M*usic *A*nalysis *P*ackage). Its volcabulary is slightly different but the analytic operations are common to both.

I would like to acknowledge the Instruction and Research Computer Center of The Ohio State University for computer time used in the development of *MAP* and *SLAM*. ■

The Digital Computer: Orchestra or Composer's Assistant?



by Arthur Layzer

There are two distinct uses of the digital computer in music today: the first is to help write a score that can be played by either artificial means or by ordinary instrumental performers; the second is to actually synthesize the musical sound from a score-like specification without the intermediary of an orchestra or conventional sound studio equipment. After ten or fifteen years of exposure to these uses of the computer, most people still regard them as rather exotic.

These applications of the computer are in fact like separate magical tricks. The production of sound by the computer out of numerical specifications may be compared to a sleight of hand trick. The trick can be explained. Once it is explained there is no more magic and everything seems straightforward. The programming of a composition on the other hand is a trick of a very different kind. The difficulty is in understanding how something of aesthetic value can be created from such drab materials as algebraic transformations, random choices and Fortran do-loops. Perhaps, in fact the trick need have no rational explanation. It might be an illusion, like the Indian rope trick. After all, the aesthetic experience itself may be completely an illusion. We of course all hope that it is a benign one. If the aesthetic experience is an illusion then presumably it is enough to demand of a programmed composition that it produce the illusion in some listeners. Now, it is known that some programmed compositions have created the illusion of art among some people. The body of experience in this area is however still not large and I concede that there are serious questions of principle. To be on solid ground I shall talk mostly about the sleight of hand trick — the synthesis of musical sound by the computer.

If you analyze the production of sound by a conventional orchestral instrument like a saxophone for example, if you analyze it from a rather strange perspective you might realize that two functions are combined in one piece of hardware: the actual physical generation of sound waves by the air resonances inside the saxophone and the structuring of the air column by the holes and wall-shape of the saxophone which determines these resonances. Of course, I count the reed, the air and the saxophonist's fingers as part of this hardware.

In an electronic studio producing tape-music the situation is more or less similar from this point of view. There is a collection of oscillators with dials. The outputs of the oscillators are mixed together or serve as controlling voltages for other oscillators. The end result is a voltage output fed to the input of a tape recorder. Inside the tape recorder the electrical wave form is copied or translated into magnetic impressions on the tape. The tape is a stored record of the electrical wave-form and on playback the magnetic impressions on the tape are again rendered into electrical impulses which this time drive speakers which by their mechanical vibrations produce the final sound wave.

Compared to the saxophone, there is the addition of various copying and translating stages. But just as in the saxophone the structuring that determines the musical properties of the final sound wave is embedded in the hardware, in this case the hardware of electrical circuits and the associated dials.

In the digital computer synthesis of music, the structuring inherent in the musical sound is completely separated from the hardware implementation associated with the final sound wave. The structuring is achieved by mathematically schematizing the relevant aspects of electronic oscillators and their coupling through Fortran-like programming and then outputting a sequence of numbers on a digital tape. The sequence of numbers is to be interpreted as a uniform time-sampling, at a certain rate, of the amplitude of the final sound wave, conveniently normalized. The digital tape is later scanned at a speed that corresponds to the assumed sampling rate by a standard piece of apparatus called a digital to analogue or D to A converter, which translates the sequence of numbers into a timed sequence of electrical impulses which are then fed into the input of an ordinary tape recorder. The rest of the process is then similar to the electronic studio case.

Historically, the first successful and comprehensive music synthesizing design along these lines was carried out by Max Mathews and Joan Miller at Bell Laboratories in the early 60's. This was a remarkable example of technical ingenuity that combined programming art with an appreciation of the electronic engineering aspects involved in obtaining a digitally structured counterpart to studio hardware equipment. Their first viable program was known as Music IV. A rather large number of improved modifications have appeared at various institutions across the country since that time. In addition, there has appeared a Music V version created at Bell Labs in 1968, this time by Mathews, Moore, Miller and Risset. ■

A Personal Statement

Though long interested in music and a good amateur clarinetist, I stayed away from computing for as long as possible, to my later regret. My major background is in theoretical physics and quantum mechanics, fields in which I'm still active at Stevens Institute of Technology. In 1967 I became seriously interested in computer music and audited a course at Princeton taught by Godfrey Winham. In recent years I have used the computing facilities at Bell Laboratories, Murray Hill where I am a resident visitor. Still more recently I became interested in computer animated poetry. The film "Morning Elevator" has been shown at a number of national conferences in music, literature and the arts, including the international conference on Computers in the Humanities at the University of Minnesota, July 1973. I've been indispensably aided by the programming skills of computer scientists at Bell Labs. Dr. Joan E. Miller in particular helped me write the visual program for "Morning Elevator" and in preliminary simulation with a minicomputer (DDP 24) before I ran the film off on the large GE (now Honeywell) batch system for handling movies, which used an electron-beam technique (Stromberg-Carlson) for exposing the 16mm film.

Readers of *Creative Computing* wishing to correspond further with Prof. Layzer, can write him at 161 W. 75th St., New York, N.Y. 10023.

BOTTOM-UP BIZET

REFLECTIONS ON IMPLEMENTING RELEASE 234.5 OF THE PEARL FISHERS

Robert P. Taylor

INTRODUCTION

Computers are wonderful devices and with them we accomplish wonderful, even astonishing things. But what astonishes me most is the freshness which computing provides into what are essentially non-computing activities. By identifying parallels with computing in a non-computing activity, I can often deepen my appreciation and understanding of a familiar human enterprise, enriching my life considerably in the process. I sometimes feel this may be a more significant reason for getting involved with computing than is the whole business of getting the computer to perform as a marvelously powerful and flexible tool in any of a host of scientific and commercial enterprises. I do not feel most computing professionals take seriously enough the importance of this "fringe benefit" of computing. In fact, I believe if we systematically encouraged and publicized the application of such insights to significant cultural enterprises, we would both enrich our culture and take a significant step toward countering the growing popular misconception of computing as a mechanistic, dehumanizing force in our society. The objective of this paper is to illustrate how this can be done in terms of one well-established and sanctified art form in our culture, grand opera. It should adequately suggest the merits of the idea.

Following this introduction, the remaining text of this paper is organized into five parts. *PART ONE* reviews the origin of the paper and likens the production of opera generally, and George Bizet's *The Pearl Fishers* in particular, to the implementation of a software system. *PART TWO* likens the New York Lyric Opera Company (NYLOC), as it

was organized to produce the *Pearl Fishers,* to a software implementation team. *PART THREE* examines the "documentation" normally available for implementing the *Pearl Fishers* system and finds that it relates almost exclusively to the audio systems of the overall opera system. *PART FOUR,* using the example of the lumination system, discusses why and how all the video systems in the *Pearl Fishers* implementation project developed their own temporary, makeshift documentation. *PART FIVE* draws several conclusions, some about opera as system, some about the fruitfulness of extending the approach taken in this paper.

PART ONE: DRAWING THE ANALOGY BETWEEN OPERA PRODUCTION AND SYSTEM IMPLEMENTATION

I first became interested in the parallels between opera and systems while singing in a recent New York Lyric Opera Company (NYLOC) production of *Don Giovanni.* As we rehearsed and subsequently performed that opera, I was increasingly impressed with the parallels between producing *Don Giovanni* and implementing a payroll or other reasonably complex software system. Many of the structural relationships between the personnel producing the opera closely resembled those characteristic of a good software project team. Activities were modularized and were developed, tested, modified and integrated just as the sub-components of a software system frequently are. Success in producing the opera seemed to depend heavily upon fitting the different strands together in the right place, at the right moment, much as the successful implementation of a software system depends heavily upon interface definition and creation.

Long before opening night, I had begun to entertain myself in slack moments by trying to look at the opera production as though it were a software system implementation. The audience became end users. Cues became interfaces. Lighting and sets became sub-systems. Section rehearsals became module testing. Rehearsals became debugging sessions. The dress rehearsal became a pilot run. The voltage limitation in the lighting power source became a hardware constraint. And on it went, until by the final curtain, I had come to see that whole NYLOC production as merely the most recent implementation of the *Don Giovanni* system.

It happened that I was teaching a course on systems analysis shortly after this experience with *Don Giovanni* and I began to think that attending an opera rehearsal might be a beneficial experience for the students in that class. After a few weeks of introduction to systems concepts and an initial experience with the software system development process, I felt the students might profit from the chance to try applying these same concepts in a foreign context. There, because of the contextual freshness, the concepts might emerge with greater clarity and the students return to the traditional systems of the course with both a better understanding of systems concepts and a wider appreciation of their grander implications. I suggested the idea to the class and they decided it would be worth trying.

Experience with *Don Giovanni* strongly suggested that one opera would be just as good as another for this sort of experience. NYLOC was, that term, readying Bizet's *The Pearl Fishers* for production so an evening dress rehearsal which coincided with the systems class hour was selected for the class's "night at the opera." Each student was given the same assignment — to attend the rehearsal and to write up at least one analogy which he or she discovers between opera "implementation" and the systems work being studied in the course.

To prepare for the trip, the class was given certain written and printed material concerning the opera, was required to listen to a recording of the latter half of Act II, and was presented with a brief outline of the organizational structure of the New York Lyric Opera Company. The materials and recording were focused on a dramatic climax in Act II which involved extensive interaction of all the different personnel involved in the opera production.

PART TWO: THE NEW YORK LYRIC OPERA COMPANY AS A PROJECT TEAM

The organization of NYLOC is similar to the organization of a software project team. Figure 2 shows the organization of the production personnel as a project team. Though soloists, their respective vocal coaches, and individual singers and instrumentalists have been omitted to keep the size of the chart manageable, the main structure is clear. There is a project leader with overall responsibility (NYLOC General Director). Two sub-system leaders report to the project leader: (1) the audio systems supervisor and (2) the video systems supervisor (the conductor and director, respectively). The first has responsibility for everything the user (audience) hears, the second, for everything the user sees. Each of these sub-system leaders have both individual and lower sub-system leaders reporting to them. And, in typical project fashion, each also assumes direct management for at least one sub-system.

While several of the sub-project or sub-system personnel under one or the other of these two sub-project leaders have still other personnel reporting to them, several others do not. For example, the Group Vocal Systems Supervisor manages all the chorus personnel and is responsible for the development of the entire choral sub-system. On the other hand, the Lumination System is a one person operation. This variety in responsibilities and in numbers of upward-reporting personnel in each case is a typical project phenomena, depending on typical project realities — size of project and budget of project. In fact, the tyranny of schedules tends to make opera companies ideal models for project teams in at least one major sense — the opera team *must* come up with an implemented system *within budget* and *on time.*

Responsibilities are initially delegated to the various sub-project leaders with enough general discussion of interface

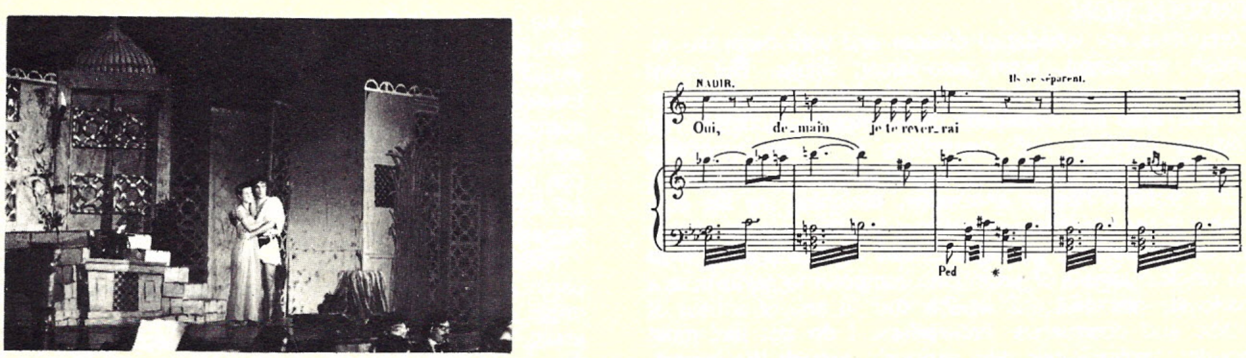


Figure 1: Sample video output and vocal source code

details to enable them all to go ahead with their individual tasks. Early rehearsals are then devoted to module or sub-system design, testing and debugging. Final rehearsals are reserved for integrating the modules and sub-systems and thus resemble full-system testing. In this respect, the implementation of the opera is a sort of bottom-up process. On the other hand, the very early rough definition of interfaces between the various sub-systems and the constant dependence on cues and stubs as modules are developed constitute a sort of top-down process. Thus, like most system projects, opera implementation involves both top-down and bottom-up approaches.

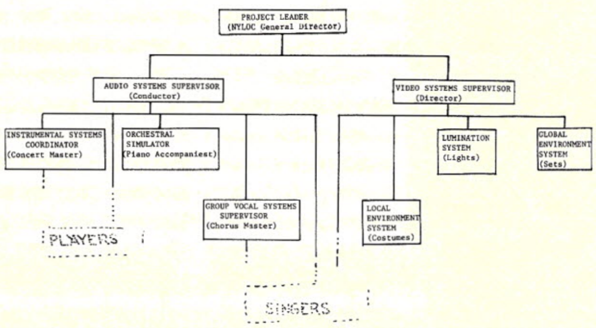


Figure 2: NYLOC Project Team

PART THREE: "DOCUMENTATION" FOR THE PEARL FISHERS

The documentation presented here is but a minute sample from the hundreds of pages existing for the *Pearl Fishers* system. For the systems analysis class, excerpts of four kinds of documentation were presented: (1) *overview* (plot summary), (2) *narrative description* (libretto), (3) *audio systems vocal source code listing* (vocal score), and (4) *user audio output sample* (phonograph recording of appropriate segments of system). Obviously no example of (4) can be included in a paper, so none will be represented or discussed here. And, though an example of (3) is shown as an opening illustration in Figure 1, this section will use the more complete *audio systems master source code listing* which includes the vocal code and much more. (The master code would have been used in class but it was unobtainable at the time.) In addition to discussing the same aspects of documentation here as in the class, this section will derive some unity from relating all examples to a single time-slice from the opera's execution.

This time-slice is underscored in the *overview* excerpt presented in Figure 3, an excerpt which adequately establishes the context for this time-slice. The systems class studied several other time-slices and their respective documentation as well, but space does not permit, nor necessity require more than this one time-slice to be examined here. The opening illustration for this article, for example, presents *video output* and an excerpt from the *audio systems vocal source code listing* relating to that time slice when the lovers embrace. It should be noted, though, that these other time slices would underscore what tremendous changes in the level of system activity occur as the act moves from the moment of the lover's embrace to a conclusion. The system must shift from an intimate, dimly-lit love duet to a furious, full-company climax with every character in the opera on stage and singing at full voice, with the orchestra playing at full strength, and with the lumination system simulating a lightning storm.

Act II *ruins of a temple.* Nurabad, the high priest, installs Leila in her position as priestess of the tribe. He tells her that she must remain in silent watch and prayer throughout the night. She is fearful of the forest sounds, but promises. Nurabad departs. As Leila trembles at the roar of wild beasts, she is suddenly reassured by the sound of a human voice. It is Nadir singing to her in the distance. She answers, and Nadir, overjoyed, tells her of his love. They embrance, but are surprised by the high priest, who has been in hiding. He calls the people together telling them that their priestess has been false to her vows. The tribesmen are ready to slay her, but Nadir shields her with his body. *Zurga, in order to protect his friend, commands the pearl fishers to disperse.* Norabad tears away Leila's veil, and Zurga then recognizes her as the same woman over whom he and Nadir had formerly quarreled. A storm arises and the people pray to the gods while the priests lead Leila away. Nadir is sentenced to death.

Figure 3: Overview documentation

The sample *video output* from a full-system test run in Figure 4 clearly shows the peak of system activity as it is being halted by Zurga's entrance. This is a critical moment for interaction of the various sub-systems. The extent to which documentation for the system details the interfacing required to implement such interaction can be determined from examining appropriate excerpts from that documentation. The appropriate segment from the *narrative description* (libretto) is shown in Figure 5. The *audio systems violin I source code* listing (concert master's part score) is shown in Figure 6. And the *audio systems master source code listing* is shown in Figure 7. Since vocal code is included in the master listing, since space is at a premium, and since the opening illustration, Figure 1, includes a typical example of vocal source code, none is presented in this section. As Figure 1 shows, no information on interfacing beyond that also carried in the master code is carried in the vocal code. Its unique component, rather than interface information generally, is the piano code which can be used during test runs to simulate or "stub" in for the orchestra.

At the point where Zurga commands a stop *(Arretez)* to the villagers' frenzied desire to slay the guilty couple, significant changes must occur.

The music must change from frenzied chorus to dramatic and isolated solo command. The activity on stage must virtually halt. The lighting and orchestra sub-systems must create a sharp change in mode. Figure 6 and Figure 7 demonstrate that the documentation carries extensive interface

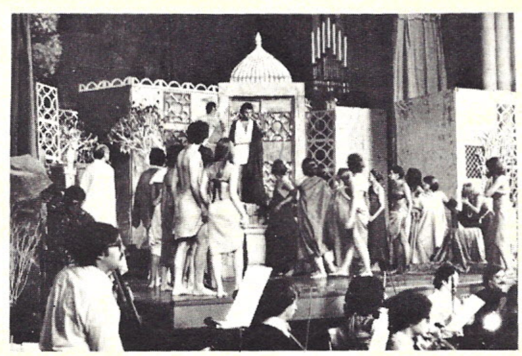


Figure 4: User video output

LEILA. Protege nous!	LEILA. Protect us!
NADIR. Venez, je vous attends!	NADIR. Come, I am waiting!
CHOEUR. Oui, pour tous deux la mort!	CHORUS. Yes, for both of them death!
ZURGA. Arretez, arretez C'est a moi d'ordonner de leur sort.	ZURGA. Stop, stop! It is for me to command their fate.

Figure 5: Narrative Description

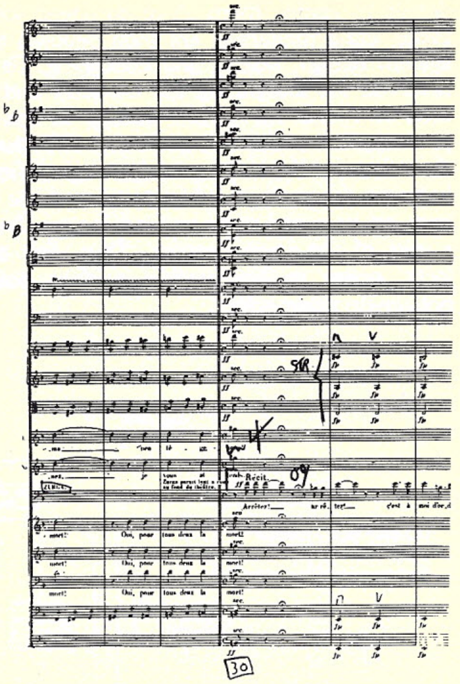


Figure 7: Audio systems master source code excerpt

detail for the audio system. The exact words of each singer, the exact pitches and rhythms of each audio system performer's notes are clearly specified. The dramatic change in mood is specified by the change in tempo at the double bar line in both source code listings' by the specification that every instrument and voice sound at full strength, once and only once, at the change of tempo; and by the specification that Zurga execute six notes in grand isolation immediately thereafter, while the instrumental sub-system remains in a wait state. The exactness of this interface between audio sub-systems is emphasized by the handwritten additions to documentation visible in both Figure 6 and Figure 7.

However, audio sub-systems interaction is only one form of interaction in the *Pearl Fishers* system. This scene certainly presupposes many decisions about interfaces between lighting, stage settings, and singer movements. Yet the documentation presented contains little or no specifications regarding such interfaces. We do find action specification stubs such as *("Surga parait tout a coup au fond du theatre"),* in Figure 7, but these are little more than can be readily inferred from the larger context provided by the singer's words. What the project team might most like to know is not even mentioned. What are the villagers to do when Zurga makes his dramatic entrance? What stage setting would maximize the impact of the entrance? Where should the guilty couple be located when the entrance begins and where when it ends? How should the lumination change during this critical moment? Not one of these questions is resolved in any way by the "Zurga appears suddenly at stage rear."

Thus formal documentation for the system carries considerable detail concerning the audio system but little concerning anything else. This is particularly noteworthy because the documentation is so extensive. The *audio systems master source code listing* alone runs to over 300 pages for the *Pearl Fishers;* the *audio systems vocal source code listing* to over 200 pages; the *narrative description* to over 30 pages; and the *audio systems instrumental source code* to over 15 volumes of 30 to 40 pages each!

PART FOUR: MAKE-SHIFT DOCUMENTATION/ A PRODUCT OF CUSTOMIZATION

In complete contrast to documentation for the audio system, that for video is make-shift and varies widely from sub-system to sub-system. Each video sub-system must go through a whole process of obtaining interface definitions; of developing tailor-made documentation concerning these definitions and anything else requiring common understanding across sub-systems; and of using the documentation to implement its sub-system (note: this documentation may not even be written down at all). Despite the differences between sub-systems, the salient elements in this process may be grasped by looking at the example of any single, particular sub-system. Such an example also illustrates why such documentation remains informal and why the collection of video sub-systems are custom components in every system which NYLOC implements. The remainder of this section is therefore devoted to the example of the lumination sub-system in the NYLOC *Pearl Fishers* implementation.

The lumination system manager had 18 lights to use. One could be used as a hand-held spot but all the others were mounted high out of reach and, once adjusted for direction, could not be re-targeted without manually accessing them from a ladder. Though the aim could not be dynamically altered during a run of the system, the brightness of each light could be, since each was connected to a separate dial on the lumination system dimmer panel.

Using this hardware, the lumination system manager based her creation of the *Pearl Fishers* lumination on four things: (1) early conversations with the video systems supervisor about his overall conceptualization of video output from the system; (2) her own recollections of work on an earlier implementation of the *Pearl Fishers* at a different site; (3) open-time opportunities to test each light in the system; and (4) observation of preliminary test runs of this version of the system.

She used (1), (2), and (4) to create a catalogue of interface points where lumination would help to define the interface both for other sub-systems and for the users. It consisted of

Figure 6: Audio systems violin I source code excerpt

(22) Zorga Ent — upper platf
up a tad

Figure 8: Preliminary lumination catalogue entry

34 entries describing the action occurring at the interface moment. The entry for the dramatic moment discussed in the previous section is reproduced in Figure 8.

She used (3) to number each light and create a table which would show, opposite the number for each light, the target, color, and other salient characteristics of that light. She then used this table and her catalogue during a final full-system test to create a final catalogue of working interface descriptions for lumination. As with the preliminary catalogue, this final one had 34 entries. This one, though, showed the exact lights to be used for each interface and the exact brightness for each light. Entry 22 for this final catalogue is reproduced in Figure 9. In that figure, the single digit numbers on the left are light identification, the double digit numbers on the right are brightness specifications.

It should be clear from this example just how customized this portion of the overall system is. The physical constraints alone would vary considerably from one implementation to another, rendering any but the most general interface documentation on lighting useless. Moreover, as the examples from the audio system in the previous section showed, there is no guidance as to where any of the singers should be on stage at any particular time nor any exact specification of the sets which must provide the on-stage context for these singers. Since all these would enter into decisions about which lights should be aimed where and when, no specifications of lumination could be meaningful without such details about the other video sub-systems.

PART FIVE: CONCLUSIONS

The discussion of the New York Lyric Opera Company as a project team suggested that, like software implementation project, the production of an opera involves a mixture of approaches, including both top-down and bottom-up procedures. The discussion of documentation used in NYLOC's implementation of Bizet's *Pearl Fishers* suggested that the opera implemented is a system which depends on both off-the-shelf and custom components. It also suggested that the distribution of these two component types very much followed the pattern suggested in the project analogy — off-the-shelf components went into implementing the audio system and custom components went into implementing the video system. This pattern existed because the audio system depended on standard hardware (voices and instruments capable of standard output) while the video system had to be built upon variables which changed with every implementation of the system.

What does all this suggest about opera and computing? Certainly that the two activities have far more in common than popular stereotypes of either would imply. And this paper has examined only a few parallels — there are many more worth looking at. Long range planning, debugging, backup procedures, and iterative problem solving would all be interesting and would probably suggest further similarity between the two activities. Those who like opera and like computing may wish to explore some of these other parallels.

What about the wider implications? For those who find opera of little interest, the basic approach discussed in this paper may be applied to other human activities. For example, the systems analysis class undertook other assignments as well as the *Pearl Fishers* one. They looked for parallels to system concepts in Defoe's *Robinson Crusoe* and turned up some very interesting ones. Then they went further afield and looked for analogies in an activity of their own choice. The papers they produced in that final effort suggest the rich lode waiting to be mined. They found parallels to computing systems work in such diverse enterprises as: incubating chicks, giving birth to a child, learning to ski, playing a season of football, and preparing a family-reunion Thanksgiving dinner.

Finally, whatever else one can say about this sort of pursuit of computing's fringe benefits, three things are clear. First, successfully calling attention to parallels between computing and other significant human enterprises should weaken the popular misconception that "computer people" engage in some sort of esoteric, mechanistic enterprise, whose methodologies have no analogue in other human activities. Second, it should also demonstrate one of computing's most significant contributions to general education — it rewards the discovery of previously concealed similarities and relationships. Third, the process of looking for such parallels is both instructive and just plain *fun!*

Bibliographical notes

Since this system is not implemented often, documentation is sometimes difficult to locate. Since the opera is old and not frequently produced, copyrights have lapsed and pirated editions have been produced. Sample audio output is also limited. Libretti in various forms are available with recordings or in libraries. They too tend to have no coyright and in some cases no extent publisher. If you wish to see a score, look for the Choudens edition under the title *Les Pecheurs de Perles* or the Kalmus version under the title *The Pearl Fishers.* ■

(22) Zorga Ent
1
2 - 80
3 - 70
4 - 50
5 - 50
6 - 50

Figure 9: Final lumination catalogue entry

This is the first in a series of articles about strategies or approaches for solving practical problems on the computer. Readers will find the heuristics and rules of thumb discussed in these articles are independent of subject matter and of great value in solving all types of programming problems from simple to very complex. —DHA.

THINKING STRATEGIES WITH THE COMPUTER: INFERENCE

Donald T. Piele and Larry E. Wood*

Experience in solving problems and experience in watching other people solve problems must be the basis on which heuristic is built.

G. Polya

Some 32 years ago, in 1945, George Polya published a little book called *How To Solve It.* Judging from the title, one might expect to find inside special techniques and sure-fire algorithms that guarantee solutions to specific problems. But this is not what Polya's book is about. Instead, it is packed full of ideas and 'rules of thumb' that are useful in attacking any type of problem but do not guarantee a solution to any specific one. Polya's methods, which he labels heuristic, are derived from the experience of good problem solvers and are characterized by their generality, their independence of subject matter, and their common sense.

Inspired by the work of Polya and recent advances in the field of artificial intelligence (e.g. Newell and Simon, 1972), Wayne Wickelgren published a similar book, *How To Solve Problems* in 1974. This book contains detailed explanations of several general problem solving strategies along with puzzles and games to illustrate each strategy. Puzzles are well suited for the task because they require the same logical thinking processes as problems in any subject area but they do not require any special knowledge. Their only drawback is that people sometimes refuse to take them seriously. They fail to see any connection between the thinking skills needed to solve a frivolous puzzle and those needed to solve more practical problems.

Recently, we have been studying the problem solving strategies of Polya and Wickelgren and have been extremely impressed with their generality and power. In preparation for a course on Thinking Strategies at UW-Parkside, we have collected many examples of puzzles and games from the pages of such classic works as The Moscow Puzzles, and Mathematical Puzzles of Sam Lloyd for use in practicing each strategy. Now we are exploring ways in which computer programming can be incorporated with these skills to solve even more complex problems. We would like to share some of our ideas in a series of articles for *Creative Computing* demonstrating the added power of heuristic problem solving skills when used in conjunction with the computer.

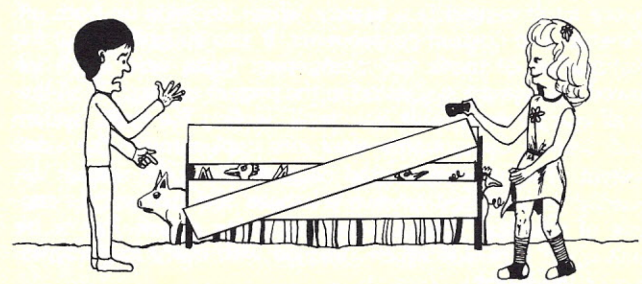
*University of Wisconsin-Parkside, Kenosha, Wisconsin 53140

Inference

In this first article, we will discuss the strategy of *inference. Webster's New Collegiate Dictionary* defines inference as "a logical conclusion from given data or premises, a judgment derived by reasoning or implication." As a heuristic problem solving tool, inference is more broadly defined to include meanings such as explicitly stating information that is implicit in the problem, making deductions and inductions, and generating and testing hypotheses. Viewed in this expanded sense, inference becomes a basic component of most problems. Indeed, it is difficult to imagine how any problem could be solved without it. As an example, consider the following problem.

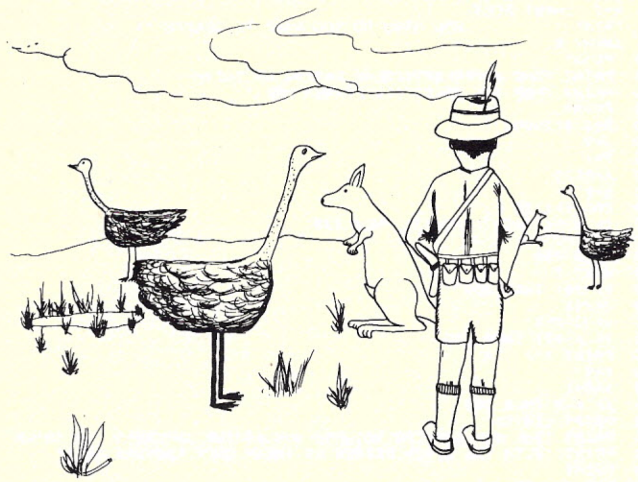
PIGS AND CHICKENS

A boy and his sister visited a farm where they saw a pen filled with pigs and chickens. When they returned home, the boy observed that there were 18 animals in all, and his sister reported that she had counted a total of 50 legs. How many pigs were there in the pen?



The first step in solving any problem is to fully understand what is implied in the problem as well as what is explicitly stated. For example, in the pigs and chickens problem it is assumed one knows that pigs have four legs and chickens have only two. This may seem trivial, but if the problem were posed with kangaroos and emus, the same inference

might be obvious only to an Australian problem solver. The next step is to deduce relationships that may exist between elements of the problem. For example, the total of 50 legs is equal to the number of pigs times four plus the number of chickens times two. Also, the pigs plus the chickens equals 18 animals. At this stage, anyone with a knowledge of algebra would probably symbolize the above relationships in two algebraic equations (e.g., $P+C=18$ and $4P+2C=50$) and solve for the number of pigs. This is certainly a familiar way to solve story problems, but for the purpose of emphasizing the usefulness of inference let's see how an Australian problem-solver might attack the same problem if it were posed with kangaroos and emus.



One usually visualizes a kangaroo resting on its hind legs with its smaller front legs in the air. An emu (which resembles a large turkey) has only two legs. Thus an Australian might easily infer that with 18 kangaroos and emus, a total of 36 legs are on the ground. Since there are 50 legs in all, there must be 14 legs in the air, which belong to exactly 7 kangaroos. Such a simple solution is very unlikely to occur to someone when the problem is posed with pigs and chickens, but what is to prevent pigs from standing on their hind legs—at least in our minds? It is probably true that the less likely we are to make a particular inference, the more likely we are to label it insight. However, in this problem, it may be more appropriate to call it hind-sight!

As illustrated in the pigs and chickens problem, drawing inferences depends heavily upon prior experience. Therefore, it may be difficult to make critical inferences with complex problems or with problems from an area unfamiliar to the problem solver. To overcome this difficulty, the computer can be a very effective tool. With it, one can rapidly generate important information related to a problem, which can serve as a basis for formulating and testing hypotheses about a solution. We shall illustrate this with the following example from the field of music theory.

The Nun's Fiddle

The Greek mathematician, Pythagoras, first discovered a basic relationship between musical harmony and number. This relationship is briefly explained by Helm (1967).

> "Pluck a stretched string of any length and allow it to vibrate; it will sound a certain pitch. Allow only half of it to vibrate and the pitch will rise an octave. If two-thirds of the string vibrates, the pitch will rise a fifth above the one produced by the total length. For instance, if the total length produces C, two-thirds of the string will produce G. (The interval C-G is called a fifth because five lines and spaces on the musical staff are traversed in going from one to the other, counting C and G.) Three-fourths of the string will yield a pitch a fourth higher than the total length (F, if the total yields C) and so on. In time the fractions become more complex and the two notes represented by the resulting intervals become more dissonant if they are sounded together."

The discovery that pleasing cords correspond to exact divisions of a string by whole numbers had mystic overtones for the Pythagoreans. They inferred that if nature and number corresponded harmoniously in music it must be true that a single order, expressible in number and ratio, governed all the rhythms of nature. This led to the myth that the orbits of all heavenly bodies were related by musical intervals. "The movement of the heavens were, for them, the music of the spheres" (Bronowski, 1973). Gradually over the centuries certain ratios corresponding to musical intervals became the basis of traditional Western music. Silver (1971) explains:

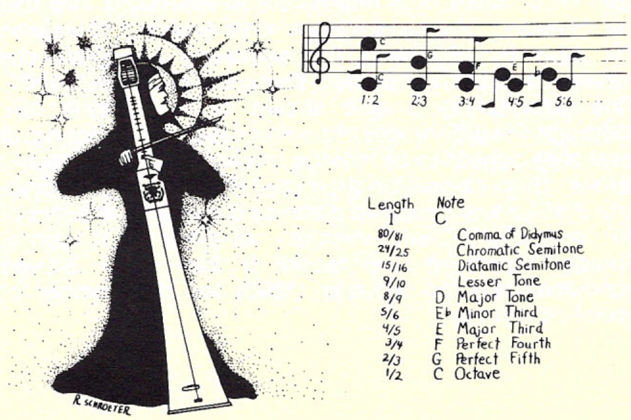
> The satisfying intervals were derived from natural harmonics, the frequencies of which are related to the natural number series 1:2:3.... Successive ratios 1:2, 2:3, 3:4...were favored. The lower ratios are pleasing; the higher ones tend to harshness and eventually become unacceptable. Certain ratios, although within the range of acceptable harshness, are regularly rejected, e.g., 6:7, 7:8, 10:11, 11:12.... There is no obvious reason for this empirical fact. However, an analysis of a large amount of material discloses that the ear prefers the following finite set called the *superparticular ratios:*

1:2 octave	8:9 major tone
2:3 perfect fifth	9:10 lesser tone
3:4 perfect fourth	15:16 diatonic semitone
4:5 major third	24:25 chromatic semitone
5:6 minor third	80:81 comma of Didymus

The relation of music to number expressed by the superparticular ratios is very beautiful, and a complete understanding of this relationship may even convince you that it is divinely inspired. The superparticular ratios are examined with the aid of the tromba marina, a late medieval bowed instrument with a single string. The instrument was frequently used by nuns and hence the German name *Nonnengeige* or nun's fiddle.

NUN'S FIDDLE

The superparticular ratios in music are related to the prime numbers and can be defined by two simple properties. Find these properties and prove that they characterize the superparticular ratios uniquely.



The first part of this problem can be answered by writing each whole number in a superparticular ratio in terms of its prime factors, i.e. $4=\ ^2$, $6=2\cdot3$, $8=2^3$, $9=3^2$, $10=2\cdot5$, $15=3\cdot5$, $16=2^4$, $24=2^3\cdot3$, $25=5^2$, $80=2^4\cdot5$, $81=3^4$. One can infer that two properties characterize these ratios: (1) Each number is of the form $2^a3^b5^c$ where a,b,c 0, and (2) the numbers in each ratio differ by one. The difficult question is whether these two properties determine the superparticular ratios uniquely. Expressed another way, is it true that if a ratio of two whole numbers satisfies conditions (1) and (2) then it must be one of the 10 superparticular ratios? This statement is, in fact, true, and it was first proved by Stormer (1897). More recently it was re-examined by Halsey and Hewitt (1972). However, to understand the formal proof requires a considerable amount of mathematical expertize. In contrast, it is quite easy to write a computer program to generate successive numbers of the form $2^a3^b5^c$ from which a number of inferences can be made. Of course these inferences do not represent a strict proof but at least they increase one's understanding of the problem to the point where a proof may be easier to discover.

Program FIDDLE was written to do precisely that—fiddle around. It allows one to specify a set of primes p_1, p_2 ... p_k from which consecutive whole numbers are generated which have these primes as their only factors. In the sample run the primes 2,3,5 are specified, and all numbers, up to 1000, which have these primes as their only factors and which differ by one are printed out. What we observe are precisely the numbers in the superparticular ratios.

Conclusion

Our ability to make inferences in problem solving is strongly dependent upon our past experience as illustrated in the pigs and chickens problem. We can overcome this difficulty in many instances by using the computer to generate information to enrich our understanding of a variety of problems—even those which are not 'divinely inspired.'

Post Script

What are the superparticular ratios for the primes 2,3,5 and 7? The answer may surprise you! Be patient; there are 23 ratios. Also, what can you infer about superparticular ratios relative to any set of primes that does not contain 2? Perhaps we have not gone far enough, and there exists two consecutive numbers beyond 1000 of the form $2^a3^b5^c$. Because a computer is limited to calculating a finite number of cases, it is impossible to absolutely rule this out. However, one can generate more evidence to weaken the case by observing the sequence of differences between successive numbers that are of the form $2^a3^b5^c$. This information is also shown in the sample run. Although the differences are not constantly increasing, at least they appear to be moving toward higher and higher values. This information prompted the authors to conjecture that for any specified distance d, successive terms of the form $2^a3^b5^c$ will eventually all differ by at least d. Shortly thereafter we found it had been proved mathematically by Stormer (1898) for any number of specified primes.

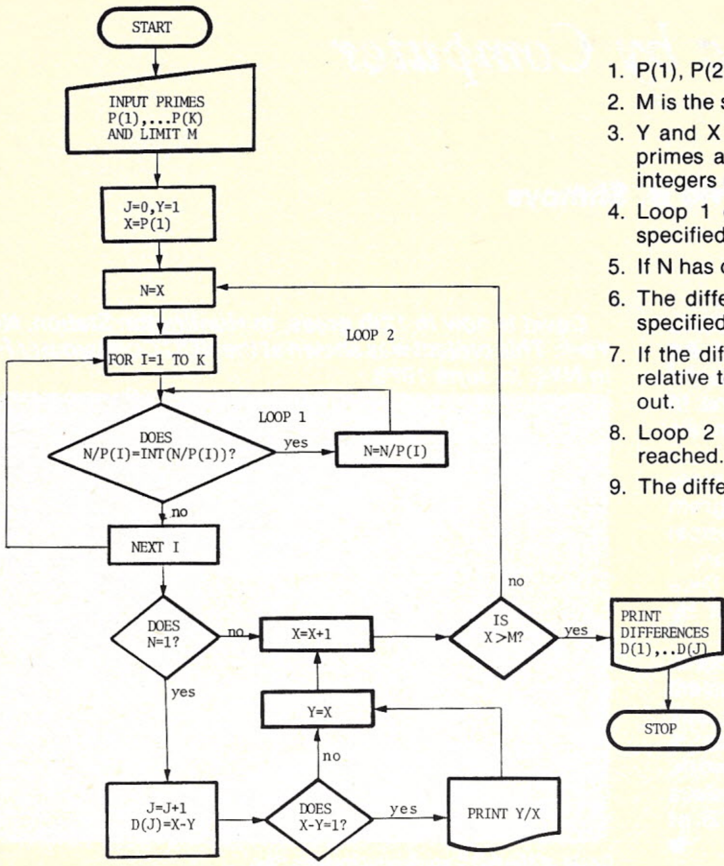
FIDDLE PROGRAM

```
LIST
FIDDLE

10  PRINT "THIS PROGRAM CAN BE USED TO STUDY SUPERPARTICULAR"
20  PRINT "RATIOS RELATIVE TO ANY SPECIFIED SET OF PRIMES."
30  PRINT
40  PRINT "  HOW MANY PRIMES DO YOU WANT TO SPECIFY";
50  INPUT K
60  PRINT "WHICH ONES ARE THEY (SMALLEST ONE FIRST)";
70  MAT  INPUT P[K]
80  PRINT "          HOW HIGH DO YOU WANT TO SEARCH ";
90  INPUT M
100  PRINT
110  PRINT "THE SUPERPARTICULAR RATIOS UP TO";M
120  PRINT "FOR THE SPECIFIED PRIMES ARE:"
130  PRINT
140  DIM D[1000]
150  J=0
160  Y=1
170  X=P[1]
180  N=X
190  FOR I=1 TO K
200  IF N/P[I]#INT(N/P[I]) THEN 230
210  N=N/P[I]
220  GOTO 200
230  NEXT I
240  IF N#1 THEN 300
250  J=J+1
260  D[J]=X-Y
270  IF X-Y#1 THEN 290
280  PRINT Y"/  "X
290  Y=X
300  X=X+1
310  IF X<M THEN 180
320  PRINT LIN(2)
330  PRINT "THE DIFFERENCES BETWEEN SUCCESSIVE INTEGERS  UP TO";M
340  PRINT "WITH THE GIVEN PRIMES AS THEIR ONLY FACTORS ARE:"
350  PRINT
360  FOR I=1 TO J
370  PRINT D[I];
380  NEXT I
390  PRINT LIN(2)
400  END
```

SAMPLE RUN

```
THIS PROGRAM CAN BE USED TO STUDY SUPERPARTICULAR
RATIOS RELATIVE TO ANY SPECIFIED SET OF PRIMES.

  HOW MANY PRIMES DO YOU WANT TO SPECIFY?3
WHICH ONES ARE THEY (SMALLEST ONE FIRST)?2,3,5
          HOW HIGH DO YOU WANT TO SEARCH ?1000

THE SUPERPARTICULAR RATIOS UP TO 1000
FOR THE SPECIFIED PRIMES ARE:

 1    /   2
 2    /   3
 3    /   4
 4    /   5
 5    /   6
 8    /   9
 9    /   10
 15   /   16
 24   /   25
 80   /   81


THE DIFFERENCES BETWEEN SUCCESSIVE INTEGERS  UP TO 1000
WITH THE GIVEN PRIMES AS THEIR ONLY FACTORS ARE:

 1    1    1    1    1    2    1    1    2    3    1    2
 2    4    1    2    3    2    4    4    5    3    2    4
 6    4    8    3    5    1    9    6    4    8    12   5
 3    7    9    6    10   2    18   12   8    16   9    15
 3    7    6    14   18   12   20   4    36   15   9    16
 5    27   18   30   6    14   12   28   36   24   25   15
 8    27   45   9    21   18   32   10   54   36   60   12


DONE
```

FLOWCHART



FLOWCHART NOTES

1. P(1), P(2), ... P(K) are the K specified primes.
2. M is the specified limit up to which the search is carried out.
3. Y and X store successive integers which have the specified primes as their only factors. J counts the numbers of such integers up to M.
4. Loop 1 checks each integer N for prime factors from the specified list.
5. If N has only these primes as factors then it is stored in X.
6. The difference between successive integers which have the specified primes as their only factors is stored in D(J).
7. If the difference D(J) = 1 then Y/X is a superparticular ratio relative to the set of primes P(1), P(2), ... P(K) and is printed out.
8. Loop 2 checks consecutive integers until the limit M is reached.
9. The differences defined in 6 are printed out.

REFERENCES

Bronowski, J. *The Ascent of Man,* Little, Brown and Company, 1973.

Halsey, G.D., and Hewitt, Edwin. "More on the Superparticular Ratios in Music," *American Mathematical Monthly,* 79:1096-1100; December, 1972.

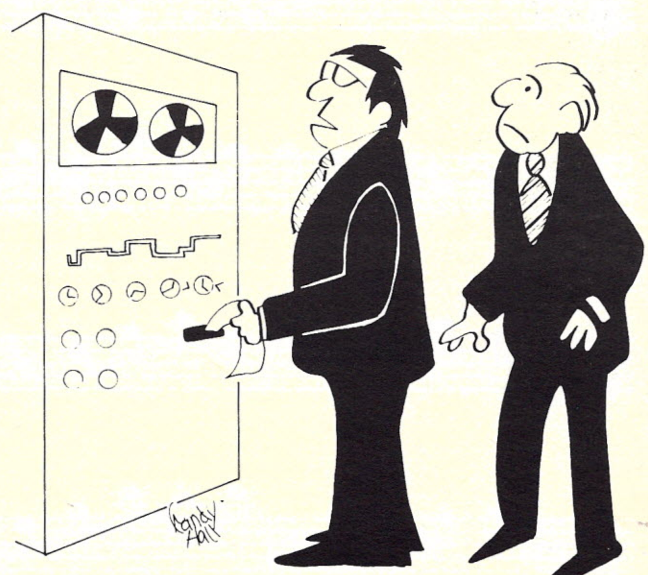
Helm, E.E. "The Vibrating String of the Pythagoreans," *Scientific American,* 217:92-103; December, 1967.

Polya, G. *How To Solve It.* Princeton University Press, 1945.

Silver, A.L. Leigh. "Musimatics or The Nun's Fiddle," *American Mathematical Monthly,* 78:351-57; April, 1971.

Stormer, Carl. "Quelques theoremes sur l'equation de Pell $x^2 - Dy^2 = 1$ et leurs applications." Skrifter Videnskabsselskabet (Christiania) I, Mat.-Naturv. K1., no. 2 (1898), 752-754.

Wickelgren, Wayne A. *How To Solve Problems.* W.H. Freeman and Company, San Francisco, 1974. (Available from *Creative Computing* Book service for $6.75 ppd.) ■



"About our prospects for that merger, it says: 'you have a snowball's chance in . . . .' "

The Transposition and Composition of Music by Computer

David B. Shmoys

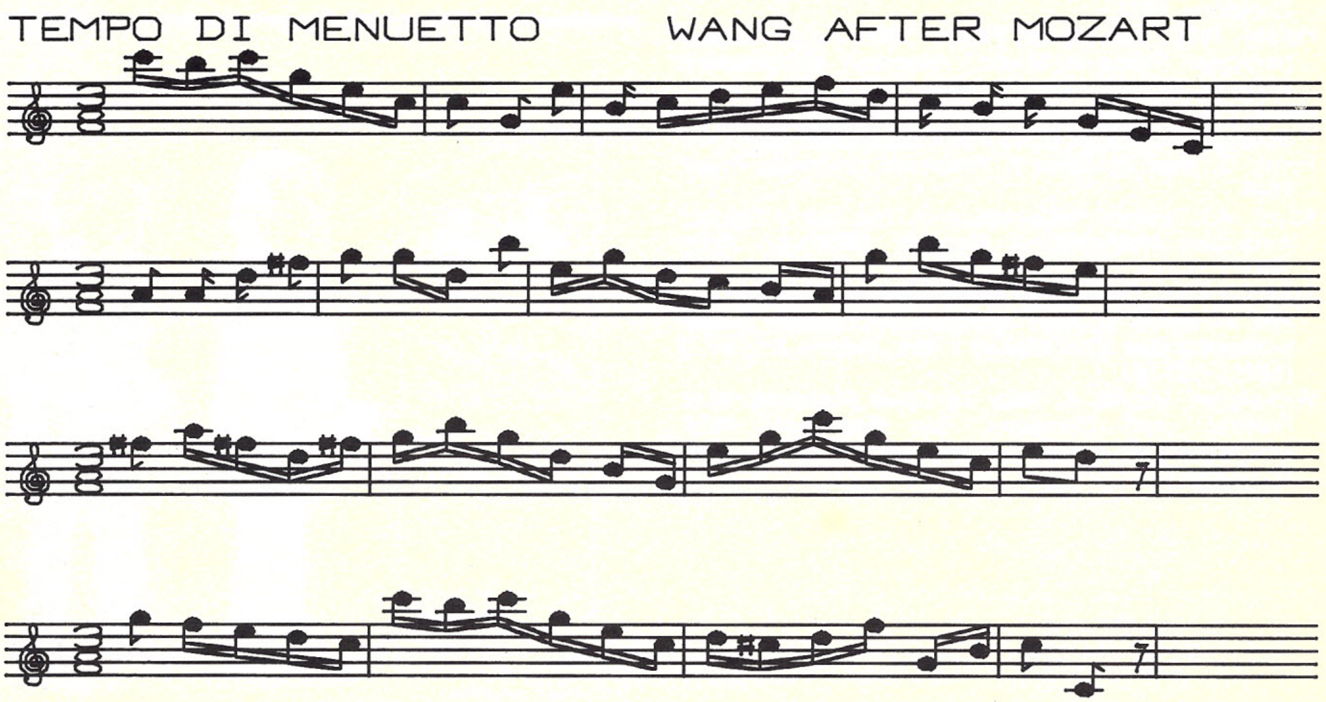
Manuscript writing and the transposition of music have long been regarded as tedious and cumbersome tasks. An efficient computer program that performs these menial chores quickly would be extremely useful to musicians. In addition, a program to get the computer to compose music that is pleasing to the ear has long been sought.

This project deals with an attempt to solve this problem using the Wang 2200 computer. I developed a program which converts alphanumber data describing musical notes into an acceptable musical score. Furthermore, I incorporated as a possible input, Mozart's "Musical Dice Game" which produces a sixteen measure minuet by random selection from a stored set of measures.

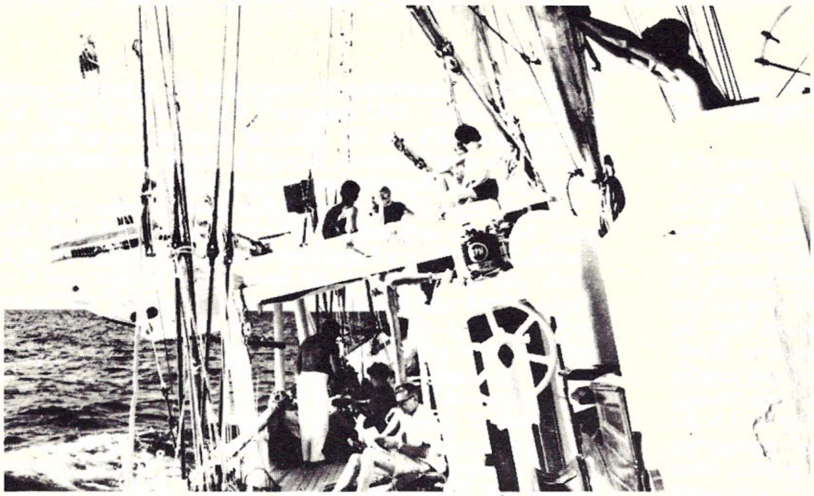
The program itself can be divided into three sections. The first section controls the drawing of the musical manuscript. The middle section transposes music any interval up or down. The final portion of the program is devoted to the composition of music using the work, "Muikalisches Wurfelspiel" written by Wolfgang Amadeus Mozart. The computer used was the Wang 2200 with 12k of core and a flatbed plotter. ■

David is now in 12th grade, in Huntington Station, New York. This project was shown at the NCC Compyouter Fair in NYC in June 1976.

David at the NCC student computer Fair.



Take off your shoes.

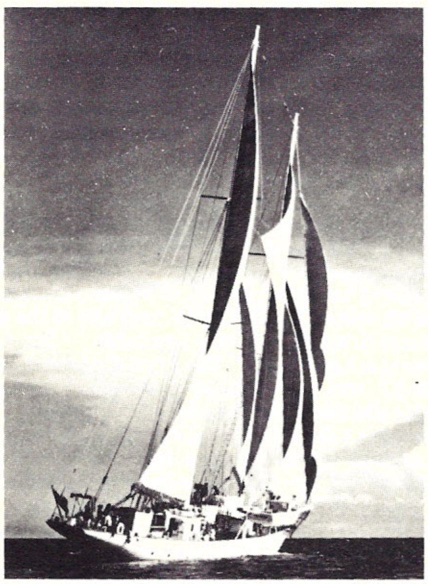


Hit the deck in shorts and a tee shirt. Or your bikini if you want.

You're on a leisurely cruise to remote islands. With names like Martinique, Grenada, Guadeloupe. Those are the ones you've heard of.

A big, beautiful sailing vessel glides from one breathtaking Caribbean jewel to another. And you're aboard, having the time of your life with an intimate group of lively, fun-loving people. Singles and couples, too. There's good food, "grog," and a few pleasant comforts...but there's little resemblance to a stay at a fancy hotel, and you'll be happy about that.

Spend six days exploring paradise and getting to know congenial people. There's no other vacation like it.



Your share from $265. A new cruise is forming now. Write Cap'n Mike for your free adventure booklet in full color.

Windjammer Cruises.

Name ______________________

Address ______________________

City ____________ State ________ Zip ________

Phone ______________________

P.O. Box 120, Dept. 17 Miami Beach, Florida 33139

THREE SYMPOSIA

by

Susan Strapac

Three Symposia. Barry S. Brook, Editor. The American Musicological Society Greater New York Chapter, City University of New York Press. 1970

The three symposia with which *Three Symposia* is concerned took place over a period of approximately one year. The first, *Musicology and the Computer I,* was held on April 10, 1965 at Rockefeller University, then Rockefeller Institute. Three speakers described their work in computer aided analysis of music, style investigation and documentation. Their lectures are published, followed by a "discussion." The second symposium was held at the IBM Systems Research Institute on May 10, 1965. This conference, *Input Languages to Represent Music (Musicology and the Computer II),* was to be a working session in which participants reported on their work in transforming musical notation into machine-readable form, after which the various methods and often conflicting ideological approaches were subjects of discussion. Most of the papers printed were submitted after the symposium, but the conference was taped and the "discussion" section includes presentations from the conference itself. The third symposium, *Musicology 1966-2000: A Practical Program,* took place at Queens College of the City University of New York on May 21, 1966. The subject was the future of musicology, not computers themselves, but, naturally, the inevitable use of computers was mentioned.

Also included are a preface by editor Brook, an Introduction dealing with the origin of the American Musicological Society, and a bibliography. Of course, because of constant rapid change in the field of computer hardware and software, state-of-the-art reports are essentially obsolete even as they are written. The editor does offer his apologies for the long delay in publishing, but it would require another book to record the changes which had taken place even by 1970 in everything with which at least the first two symposia were concerned. The lone benefit of the delay was to allow the extensive bibliography to include items published up to the middle of 1970.[1]

Some of the projects and ideas detailed in *Three Symposia* have either never gotten off the ground or have been discontinued for one reason or another. ALMA (Alphameric Language for Music Analysis), developed by Murray J. Gould and George W. Logemann and based on the Plaine and Easie Code invented by Barry S. Brook and Murray J. Gould, is described at length (P. 57-90) and in detail. Yet it has never really found users. Roland Jackson and Philip Bernzott ("A Musical Input Language and a Sample Program for Musical Analysis," p. 130-150) open their article with the statement: "Computer analysis of music opens a new and exciting area to the scholar .... Obviously, so important a research tool can no longer be ignored by the serious musicologist." Yet both have disappeared from the computer/musicology scene. Michael Kassler's music-reading machine ("An Essay toward Specification of a Music-Reading Machine," p. 151-175) has not since produced any reportable results.* However, he is presently at work in Australia under the auspices of a Guggenheim fellowship on a mathematical analog to the Schenker system of musical analysis. Eric Regener's LMT-SAM (Linear Music Transcription-System for Musical Analysis) described on p. 181-184 has gone nowhere, though he is still involved in the area of computer applications to musical analysis.

Though some ideas have failed to develop, others have borne fruit. Allen Forte's article "The Structure of Atonal Music: Practical Aspects of a Computer-Oriented Research Project," p. 10-18, has led to his recent book *The Structure of Atonal Music.*[2] Professor Forte's work exemplifies the "old" saw that having a computer is no substitute for having an idea. The computer and computer techniques are so subordinate as to be almost unapparent in the book. The computer, rather than suggest areas of inquiry, was the only logical means of handling the "accurate processing of a large amount of information, a task virtually impossible to carry out by hand methods" as the author himself says. Lewis Lockwood ("A Stylistic Investigation of the Masses of Josquin Desprez with the Aid of the Computer: A Progress Report," p. 19-27) has since dropped out of the music/computing field, though the project itself, being carried on at Princeton under Professor Arthur Mendel and Thomas Hall, is yielding excellent results using computerized variant reading comparisons in tracing the manuscript sources of Josquin's Masses.[3] Barry Brook's projections ("Music Documentation of the Future," p. 28-36) for RISM (International Repertory of Musical Sources) have come true. Appearance of the sixth volume of the "AI" Series (cataloguing all manuscripts of individual composers from the late 16th to early 19th century) is imminent and the "AII" Series will involve a computerized catalogue of well over 3,000,000 musical incipits. RILM (International Repertory of Musical Literature) — computer assisted, edited by Dr. Brook, and one of the most valuable music research tools currently available — has come into being since the three symposia. "The Ford-Columbia Input Language," p. 48-52, was summarized from tapes of Stefan Bauer-Mengelberg's discussion at the second symposium. This language, known now as DARMS, is alive and well, has some dozens of users around the world, and was one of the focal points of MUSICOMP 76, a workshop in computing and music supported by the National Endowment for the Humanities, held this past July at S.U.N.Y.-Binghamton with Mr Bauer-Mengelberg as its director. (A reference manual is currently available.[4]) DARMS is the most comprehensive of all the input languages for music, suitable for analysis, thematic indexing (for example, Harry Lincoln, a participant in the second symposium, uses DARMS for his thematic index of renaissance music), and music printing. "The Plaine and Easie Code," p. 53-56, is the language used for RISM's incipit catalog from which, with some adjustments, the incipit can be printed in elegant musical notation. Jerome Wenker ("A Computer Oriented Music Notation Including Ethnomusicological Symbols," p.91-129) is still actively at work in the area. MUSTRAN II, which is an upward compatible development of the system he described

at the second symposium, has considerable software support and has found several users, especially in the area of ethnomusicology.[5]

Naturally, to comment critically on the quality or lack of it in any of the articles in this book is uncalled for since time itself has reviewed each author's work. Some of the longest and most detailed descriptions concern essentially stillborn ideas. Some of the briefest outlines have been precursors of the greatest successes. The book's value now, and it is not inconsiderable, is as an historical document. It should be read as such, keeping in mind what of its contents has succeeded and what has failed. The reasons for failure especially are legion and impossible to report on accurately and with fairness to all sides. Probably the most "undated" material in the book is to be found in the last symposium (especially in the articles by LaRue, Palisca and Zimmerman) where machine-oriented detail was not at issue.

An idea of quality and merit will weather all storms of technological change, and re-reading of *Three Symposia* ten years after bears this out. ■

*At the second symposium, Mr. Kassler actually spoke on the IML-MIR system being used for the Josquin project at Princeton. A transcription of his comments may be found in the Discussion and Commentary section, p. 178-181.

FOOTNOTES

1. It is not, however, unique in the field. An annual bibliography on computer applications in music and musicology is published in *Computers and the Humanities,* a periodical edited at Queens College of CUNY by Joseph Raben. A compilation by their former music bibliography editor, Stefan Kostka (*A Bibliography of Computer Applications in Music (Music Indexes and Bibliographies 7*), Hackensack, N.J., 1974) includes material published into 1973.
2. Allen Forte, *The Structure of Atonal Music,* Yale University Press (New Haven:).
3. Arthur Mendel and Thomas Hall, "Princeton Computer Tools for Musical Research," *Informatique et Sciences Humaines* 19 (December 1973), p. 35-59.
4. Information on obtaining the manual, which was used at MUSICOMP 76, may be had by writing to its author, Raymond Erickson, at the Music Department, Queens College, C.U.N.Y., Flushing, N.Y. 11367.
5. Jerome Wenker, "MUSTRAN II: A Foundation for Computational Musicology," Computers in the Humanities, J. L. Mitchell, ed., University of Minnesota Press (Minneapolis: 1974), p. 267-280.

Tektronix 4051 BASIC Manuals

Four manuals and two identical pre-recorded tapes accompany the Tektronix 4051 Graphics System, whose hardware and software were reviewed in the Nov/Dec 1976 issue (page 20). One of the four is the operator's manual, covering (and expanding on) the tutorial and plotting programs on the tapes, with sections on "keys, buttons and switches," routine maintenance, and various appendixes. The second book is a reference manual.

The other two, not available at the time of writing the review, are the BASIC manuals, 8½ by 11 inches in size, almost ⅞-inch thick, in plastic bindings.

The "Introduction to Programming in BASIC" has seven chapters: Essentials of BASIC, Directives, Arrays, Character Strings, Subroutines, Extended I/O, Graphics. The seven appendixes are on hierarchy of operations, ASCII code, error messages, etc. The book covers just about everything one might want to know about using BASIC on the 4051, but it is an uneven book, with both fine and poor sections. The programs get too complicated too soon; the reader is not led up to each with enough of a firm background in the preceding pages. In many places, where examples would help immensely, they aren't given.

The first real program, which takes 13 lines to calculate the roots of a quadratic is fairly simple and well-explained. The second and fourth programs are simple, but the third is a little too complex at this stage of the game. The "example programs" at the end of each chapter are often given with little or no explanation, and are often too complicatyed for the beginner at that particular point in his learning of BASIC. This book is more like a reference manual, assuming the reader to be knowledgeable, although the introduction says this is a "fundamental approach."

There are some fine sections, such as the 4½ pages that give one of the best explanations I've ever seen of bubble-sorting, with a flowchart, pictorial, and full clarification. Also good is the section on two-dimensional arrays. Although the book is quite strait-laced, it does include what might be called a "fun program," which computes the Arabic equivalent of Roman numerals, plus a real far-out program that changes a sentence to its Pig Latin form!

There are some unique functions I hadn't seen before, and which may have been created just for the 4051, such as FUZZ which establishes the degree of precision for comparisons, and SECRET, which prevents a program from being listed.

Output formatting, in the Extended I/O chapter, gets quite complex on a 4051, with 13 operators and 7 modifiers. To print a string of -------, this manual has USING "X, 51""-""", whereas everybody else uses just PRINT "----------." Nice to have everything automated, but this means just so much more to remember, or to look up. Or maybe it just takes getting used to ....

The fourth manual, "Introduction to Graphic Programming in BASIC," has nine chapters: Graphic Statements, Data Input, Graphing, Transformations, Axis, Labels, Enhancements (dashed lines, cross-hatching), Pictures, and Three Dimensions. Of the seven appendixes, two are exactly the same as the other BASIC manual: 6-page glossary, 9-page list of error messages.

This manual also assumes a knowledge of programming, as very, very little explanation of programs is ever given. The book seems to simply show programs that are meant to be copied out, rather than trying to teach programming. However, there are, near the end of this book, in the chapter on enhancements, four programs that do get full explanations. These programs can really use the clarification, as they are complex and long (for a manual), 62 to 81 lines each. Here the book is extravagant, giving eleven ways to terminate an INPUT loop. Yet in other places it is stingy, especially with explanations and examples.

The chapter on Pictures gets quite difficult, with very long programs, of up to 100 lines, on the various types of projections (oblique, orthographic, etc.).

Here, then, are two books that present all the information, but without much imagination or examples, or explanation of programs. Curiously, the manuals seem to have been written as though the 4051 were not available to the reader. There are only two places in the manuals where the reader is asked to use the 4051. The author or authors could have written much more useful books by making them interactive, by having the reader use the machine on each page, to teach him by hands-on use. Perhaps the authors felt it best to write what are essentially reference manuals, and let the reader try out, on the 4051, whatever he feels like.

These manuals do not really teach, as other BASIC books do, since they do not progress in an orderly fashion, explaining programs as they go, building up a gradual picture of how to use the language. But there is a great deal of useful information here, apparently the whole story on the 4051, presented in a workmanlike fashion. Because of this, many people with previous experience will no doubt find the manuals more than adequate. But for the beginner, they leave something to be desired.

How to rate these manuals? For content, B or even B-plus. For style, C or perhaps C-minus. If only the manuals had been written with as much imagination as went into the design of the 4051, they would be real winners.

Stephen B. Gray

ws... reviews... revi

Reviews Editor: Peter Kugel, School of Management, Boston College, Chestnut Hill, MA 02167.

Readers: Want to be a reviewer? Write to the Reviews Editor directly. Publishers: send materials for review to the Reviews Editor.

The Computer and Music. Edited by Harry B. Lincoln. Cornell University Press. 21 articles. xvi + 354 pp.; hard cover. $17.50. 1970.

In a field that is expanding as rapidly as computer-use in music, a book published six or seven years ago may seem hopelessly out-dated. Actually, any interested person reading this book today can gain a great deal of historical and practical knowledge of the field. That is possible because the articles offered by the various contributors are written for various purposes and presume various levels of technical background.

Articles are included that are chiefly philosophical, while others are chiefly descriptive in the technical or research sense. Of the longer articles, there is one that traces the development of technology in its relation to sound and music, and another that surveys computer-composed music to 1970.

Unfortunately, computer-assisted instruction in music is left out entirely. Considering the highly developed state of CAI today, it is interesting that in 1970 the editor of *The Computer and Music* could write: "A third area, computer-assisted instruction in music theory, has emerged too recently to assess its effectiveness or possibilities" (Lincoln, 1970, p. xi).

There is no way that the encapsulated descriptions that follow can do justice to the articles themselves. The point is that, even with the exclusion of CAI, there is in this book a great variety of offerings for musicians, researchers, educators, composers, programmers, and technicians alike. It is hoped that the descriptions will give an impression of the subject with which each contributor is concerned and how the computer relates in each case.

The book is divided into six parts. Part One, *Historical Background,* contains a single twenty-page article by Edmund Bowles on the subject of the development of technology as it relates to music and sound. It is a fine article that uses language with remarkable succinctness. The use of computer-related terms to help describe music-related devices such as the hydraulic organ and the phonograph serves remarkably well.

Part Two, *Music Composition,* contains an article by Brun and an article by Strang that concern themselves with the ethics of computer composition, a fifty-four page article by Hiller that surveys music composed with computers, and an article by Citron that supplies instructions for inputting elements of music to the computer with MUSPEC (a programming language) and describes the processes and results of synthesizing computer output. It is one among several articles that are essentially descriptions of specific projects involving problems of computer input and output.

Part Three, *Analysis of Music*, contains seven articles (each under sixteen pages) all of which are descriptions of specific projects. Perhaps because of the obvious permutational aspects of harmony and the relative ease with which intervals in music can be turned to numbers, many of the projects concern themselves either with searches for the interval combinations in specific pieces of music, identification of style through interval combinations, compilation of statistics related to intervals, or analysis of intervals for the purpose of synthesizing music.

Articles by Fiore and Fuller analyze the music of Webern. Articles by Lefkoff and Stoney each present statistics related in the former case to the discovery of similar segments in the forty-eight permutations of a twelve tone row and in the latter to the problems of equal temperament. The Morton and Lofstedt article describes three Fortran programs that are coupled to numerical definitions of tonal material. Two are for composition and one for analysis of tonal music. They report: "At present, to be sure, both programs produce a conventional four-part music not unlike that achieved by music students at the end of their first year of collegiate study" (p. 161). Roland Jackson's article concludes that the most interesting observation in his study is the selection of certain harmonic colors by Webern, Stravinsky, and Varese which "... are heard frequently enough to provide a sense of unity with the piece as a whole" (p. 146). Youngblood's project, using a modification of the DARMS code, encoded music of Bartok, Schoenberg, and Hindemith for the purpose of establishing composer identification through root progression analysis.

Part Four, *Ethnomusicology,* contains two articles. Benjamin Suchoff reports upon procedures and findings obtained from computer programs that extract comparisons (such as interval sequences) from data encoded with the Ford-Columbia Representation. Pitch, duration, and other elements of music notation are encoded with this encoding system. Music used in the experiments included melodies from the folk song collections of Bartok. The Lieberman article is related to computer recognition of patterns in certain improvisations heard in Javanese gamelan music. The reviewer was impressed by the interesting and practical aspects of this study along with its implications for analyzing improvisatory styles and its slight overtone of what might be called an ethical problem. There is a thread of concern, in this book, about relinquishing decisions to the computers. As might be expected, the composer-contributors seem to address themselves to this concern more than the researcher-contributors.

Another thread that winds through the book is the search for some further definition of the concept of "style." Depending upon how well elements of style are defined, the computer ought to be able to identify composers and periods given the encoded musical elements of a piece. It ought even be able to output data that can be synthesized to produce pieces in various styles.

Part Five, *Music History and Style Analysis* contains five articles. Two of the articles are directly concerned with using the computer to further define elements of style. Crane and Fiehler report upon numerical or statistical methods of comparing musical styles. They are able to picture some of the results of their work with a dendrogram showing twenty chansons clustered according to style. "Music Style Analysis by Computer" by A. James Gabura is one of the longer articles (53 pages). Gabura is able to describe clearly the use of such concepts as "key" based on "pitch-class distribution" and "root movement," which are concepts experimented with for the purpose of developing computer-based style analysis. Gabura also describes methods for extracting such concepts or "parameters" from pitch and duration data for the purpose of identifying individual style in the pianoforte music of Haydn, Mozart, and Beethoven.

The remaining three articles constitute practical projects in the area of Music History. Barton Hudson describes a proposed catalog of French chansons inputted with the DARMS encoding system and accessible by musical incipits and other references. Earle Hultberg reports on programs that transcribe tablature to standard notation. Theodore Karp reports upon a

computer-based system for determining degrees of melodic resemblances in Notre Dame *organa dupla.*

Part Six, *Music Information Retrieval,* contains two articles. The first (26 pages) by Michael Kassler is taken up chiefly with "... a description, in the form of a programmers' manual, of a special-purpose programming language called MIR — the acronym of the phrase 'musical information retrieval'" (p. 299). The second is a detailed proposal for the development of a computer-accessible music library catalog for scores and phonorecords. The authors (Tanno, Lynn, and Roberson) are not referring to retrieval of individual notes in compositions, of course, but to literal library references such as "composer," "instrument," "subject." They make a strong case for such a system claiming not only increased efficiency, speed, and accuracy, but lower cost. Most important to them, however, are "... the myriad of information possibilities inherent in the data base being developed" (p. 342).

It is certainly outside the scope of this review to update the projects in this book. It is hoped that a new and updated edition of *The Computer and Music* will soon be forthcoming.

Robert W. Placek
Assistant Professor of Music
University of Georgia
Athens, Georgia

The Technology of Computer Music. Max V. Mathews. M.I.T. Pres, Cambridge, Mass. 188 pp. 1969

The term "computer music" has been used to describe computer-aided activity in all phases of music production. The subject of Mathews' book is not that broad. It is concerned primarily with a specific method of computer sound synthesis, the direct calculation of the sound pressure wave. I have come to refer to this method as *direct synthesis.* Other methods exist which involve a computer in control of external sound producing and sound shaping modules. These methods might be referred to collectively as *indirect synthesis.* The main advantage of direct over indirect synthesis is that, at a greater cost in CPU time, the composer is freed from the limitations imposed by the availability and configuration of the external sound modules. With indirect synthesis the number of each kind of module is finite. By dealing directly with the sound wave the composer has virtually an infinite supply of computer-simulated modules, giving him virtually infinite control of the sound. He can produce any sound he can specify and vary it over time in any way he can imagine. The trick is in being able to specify what is desired. The MUSIC V program described in this book, a product of the work of Mathews and others at Bell Labs, provides the composer with a powerful language for doing just this.

Chapter One, "Fundamentals," describes the technique for representing sound as a sequence of numbers, known as samples, and then transforming the numbers into sound. Each sample represents the amplitude of the sound wave for an instant of time. Once the samples have been generated, they can be transformed into a variable voltage which can then be amplified and used to drive a loudspeaker. These discrete samples represent a continuous function varying over time in the same way that the individual frames of movie film represent a continuously changing picture. The speed at which the individual units are presented is so fast that the observer cannot resolve them; he is aware only of continuity. Various kinds of distortion arise if the range of numbers used to represent the amplitudes is not wide enough (quantizing error) or if the duration of the sample is not short enough (sampling error). 30,000 samples per second must be calculated if the full 15,000 herz bandwidth of human hearing is to be represented. Attempting to generate sounds with partials higher than half the sampling rate results in another kind of distortion (foldover).

Chapter Two consists of a graded sequence of scores written in the MUSIC V language. Through these examples Mathews describes the operation of the program and gradually introduces various components of the language. MUSIC V is written for the composer accustomed to designing his instruments by patching together the sound modules found in conventional synthesizers. In MUSIC V the patching is done symbolically. Information for playing the instruments is given by the *notes,* data records of pitch and time specifications. Although the input format is tedious, the composer can write conversion and composition subroutines in FORTRAN or assembly language which allow the computer to take over much of the "dirty work" while the composer works with a more musically-oriented format.

Chapter Three, "MUSIC V Manual," presents much of the same information as Chapter Two but in a systematic manner, designed for easy reference.

Although the book is now seven years old and MUSIC V has undergone further modification since its publication, it still has much to offer. There are similar programs running on institutional computers all across the country. This book is a good introduction to any of them. Chapter One is a thorough guide to the sampling technique. It and the Appendix contain the mathematics required for a thorough understanding, yet the chapter is comprehensible to those who have not progressed beyond a basic understanding of algebra. The mathematical sections have been flagged with asterisks, and the rest of the chapter has been so arranged that the reader can skip the math and come away with a working knowledge. Mr. Mathews has accomplished this feat by clear and non-technical explanations supplemented with copious illustrations.

The book could be used as a text. There are lists of sample problems and bibliographies which lead the reader to some of the most significant literature on the subject wirtten before 1969. Since it is almost entirely in FORTRAN, MUSIC V is easy to implement on any system and also easy to study. Consequently, the book and the program together provide a model for anyone intending to write a sound synthesis program.

The criticism offered here is not of the book itself but of the kind of program which has been described. Direct synthesis programs are extremely powerful, but they also require large amounts of CPU time. It is quite conceivable that a given second of sound may take a full minute to compute. If the composer has access to a computer for the time required, he then must contend with the special requirements for converting the digital representation of the sound to a form that can be recorded and played back by a conventional tape recorder. Sometimes these facilities are not available at the same computer center which produced the digital tape. This can cause long delays between submittal of a job and the return of an audible product. However, advances in computer technology will probably deal with these problems before a more powerful sound generation technique is devised. In Mathews' book this technique is given a classic presentation.

Richard E. Saalfeld
Columbus, Ohio

Rogers, John E. "The Uses of Digital Computers in Electronic Music Generation," from *The Development and Practice of Electronic Music* Edited by Jon H. Appleton and Ronald C. Perera. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1975.

It is exciting to find a book on electronic music destined for wide acceptance which devotes almost one-third of its content to computer use for electronic music generation. Mr. Rogers places a deserved emphasis on the computer when he suggests that the "expansion (of electronic music studios) must involve digital computers as essential units in electronic music generation." This support of computer use might even be considered mild, since it is the reviewer's belief that, even more than as an expansion, the computer will be *the* central and basic unit in future electronic music studios, ultimately requiring their total re-design.

Rogers cites two basic methods for the computer's use: as a digital control device for an analog studio and for pure computer sound synthesis. The logical trend to systems based on digital control of digital devices is referenced, but both enthusiasm and material on this approach are lacking in this chapter of the book.

Giving an excellent recommendation to this work is easy, since it provides valuable, specialized information not frequently found in a book with such wide circulation. However, the reader must be able to move from the interpretation of general information usually understood without in-depth knowledge to the fairly detailed descriptions appealing mostly to those who have a specialized interest in computer music. The transition from general to specific is quick and apparent, with enough material included to develop each into separate and valuable chapters.

The general information covers: the limitations of present-day studios which may be eliminated through computer use; the basic uses of computers in electronic music; the major characteristics of computer technology such as timesharing, batch processing, and minicomputer applications; and computer programming techniques. In addition, a replication of information readily available and clearly presented in Max Mathews' *The Technology of Computer Music* (M.I.T. Press, 1969) is included. This fact is mentioned, not to criticize Mr. Rogers for including his own explanations of D to A conversion, foldover distortion, sampling, quantizing, and other basics inherent in digital representation of sound waves, but to emphasize the importance of Mathews' earlier work now recognized as a published landmark in the field.

Several explanatory points related to simple programming concepts were disappointing when compared to many other excellent sections of the chapter. For instance, the technique of branching, with all of its unique and valuable characteristics, was presented too simply as a mode which "allows certain instructions to be skipped or branched around." In a somewhat similar fashion, the notion that "the computer should ... be programmed in a high-level language, preferably Assembly Language" is puzzling in that much more sophisticated languages are available, even on minicomputers, for executing a series of assembly instructions in one single command. Rogers' justification for use of assembly language due to economic factors associated with its greater speed seems invalid with modern technology.

The more technical aspects of the chapter dealt with computer sound synthesis, based (too much) on the MUSIC 360 coding system of Barry Vercoe. Although the reviewer considers this as a weakness of the article, Rogers purposely limits himself and states, "MUSIC360 can be understood, at a basic level, with only the information presented in this chapter." The weakness here that is felt is one of narrow coverage of "The Uses of Digital Computers in Electronic Music Generation," since other different approaches have equal validity.

Although I would have preferred it if Mr. Rogers had looked more toward the future — to advancements that are bringing computer technology out of its infant stage, to general availability for all, and to the increased ease of use and operation, he has made a major contribution in reporting some of the continuing developments in the field.

David Swanzy
Professor and Coordinator of Graduate Studies
in Music Education
Southern Methodist University
Dallas, Texas

Computer Careers, Planning, Prerequisites, Potential. John Maniotes and James S. Quasney. Hayden Company, Inc., Rochelle Park, New Jersey 07024, 180 pp. $4.95. 1974.

The eight chapters of this book provide the reader with a sound basis for making decisions concerning a career in electronic data processing. The second chapter, titled "How Computers Do It," provides an excellent overview of computer operation written in plain language which can be understood by individuals without a background in electronic data processing.

Other chapters cover such areas as the kinds of educational training required for the variety of jobs in EDP and do an admirable job of outlining the requirements of various programs. Unfortunately, the book does not contain information on educational computing, which is rapidly expanding as a "new" enterprise. Chapter 5 provides an excellent overview of the costs of an education and how the student can defray such costs through grants, scholarships and fellowship programs. The resource list provided is extensive in terms of the agencies which assist students enrolled in EDP fields.

Chapter 6 discusses the problem of finding a computer-EDP-oriented job and presents an excellent outline of a procedure for developing a resume which can be used universally in preparing such a document. It also provides excellent hints on the interview process and the variety of tests that one may be required to take in seeking a position.

The final chapter of the book describes three types of institutions that offer degrees in electronic data processing. It gives the pros and cons of each type of institution and points to the need to look critically at all institutions to determine if the objectives of the program are in harmony with the career objectives of the individual.

Although texts typically involved in computers and computer professions are out of date within a few years of publication, this text can provide a continuing source of information for students who are graduating from high school and trying to determine which institution they should attend, as well as students who are already in higher education who are looking forward to a position in the field. The book may seem elementary to many, but this reviewer feels that it does provide an excellent resource for a younger student who is in the awesome position of trying to make a decision for a life-long career. I would also recommend the book for guidance counselors, since it is written in terms that non-EDP-oriented people can understand and apply.

Daniel Krautheim
Columbus, Ohio

Introduction to Programming Languages. W. Wesley Peterson. Prentice-Hall Inc., Englewood Cliffs, N.J. 358 pp., $12.95, 1974.

In this easy to understand book, eight programming languages are presented under the four major sections *Scientific, Data Processing, Character String Processing,* and *List Processing.* The languages used are BASIC, FORTRAN, COBOL, PL/1, ALGOL, APL, SNOBOL, and LISP. PL/1 is covered in each of the four major sections in which the language features appropriate to that section are discussed. The use of simple problems to illustrate the features of the language under discussion makes this an attractive text for an introductory course on programming languages. Some of the more advanced techniques such as recursion and list processing are clearly explained. An additional aid is the use of the same set of problems for each chapter (on language) within a major section. Some will complain about the lack of exercises for each chapter, but with the various illustrations, alternative problems should easily come to mind.

The book is a good reference work, but it is IBM orientated, and the implementation for a language will vary from one vendor to the next. Therefore one must consult the system's reference manual before using the language. I found the author's coverage of PL/1 to be more than adequate and his section on PL/1 use of based variables in list processing to be outstanding. I recommend the book as a text in a survey course on programming languages or as a reference work for the practicing programmer.

William J. Marshall
Chelmsford, Mass.

Puzzles and Problems for Fun

THE KEYRING PROBLEM

Consider a keyring with 5 keys. Because of the structure of a keyring, each key is adjacent to 2 other keys. The problem is to engrave a number on each key so that the keying possesses the following property:

For any number, n, between 1 and 21 inclusive, there exists an *adjacent* group of keys whose engraved numbers sum to n. For example: If 1-2-4-x-x is part of the keyring, here are the possibilities:

n		*keyring*
1	=	1
2	=	2
3	=	2+1
4	=	4
5	=	?
6	=	4+2
7	=	4+2+1

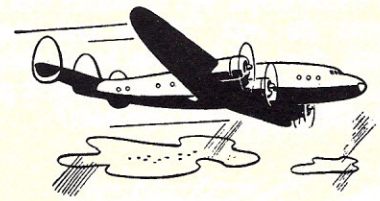
Clearly, it is not possible to form a 5 with adjacent keys. Can you write a program to solve this problem in a *reasonable* amount of time?

Rob Kobstad and Mike Lucey
Notre Dame, IN 46556

P.S. A Fortran solution on a GA 18/30 required less than 1 minute to compile/solve the problem.

THE FRIENDLY SKIES

Every hour on the hour a jet plane leaves New York for Los Angeles and, at the same instant, one leaves Los Angeles for New York. If each trip lasts exactly five hours, how many planes from L. A. will each plane from N. Y. see (assuming good visibility, of course)?



Thinkers' Corner

by Layman E. Allen © 1975

SET THEORY PUZZLES

How many of the problems (a) through (f) below can you solve by forming an expression that will name the number of cards in the universe that is listed as the GOAL? (Suppose that each letter and symbol below is imprinted on a disc.)

The expression must use:

(1) all of the discs in the REQUIRED column
(2) as many of the discs in PERMITTED as you wish, and
(3) exactly one of the discs in RESOURCES

Universe of Cards	1	2	3	4	5	6
	B	A B D	D	B D	C D	A C

Examples:

The expression A names 2 cards (2,6).

The expression A' (complement) names 4 cards (1,3,4,5).

The expression B ∩ D (intersection) names 2 cards (2,4).

The expression B ∪ D (union) names 5 cards (1,2,3,-4,5).

The expression B-D (difference) names 1 card (1).

Problem	GOAL	REQUIRED	PERMITTED	RESOURCES
(a)	4	∪	ACD∪	ABC∪ - '
(b)	3	B	CD∪∩	BD∪∩ - '
(c)	5	C∪	ABD∩	BC∪∩ - -
(d)	5	C -	BC∪∩	CDD∪∩ '
(e)	3	A∩ '	BC∪∩	ABC D∩ -
(f)	4	BD -	AD∪ -	BCD∪∩ -

If you enjoy this kind of puzzle, you might like playing ON-SETS: The Game of Set Theory. Free information about this and other instructional games is available upon request from The Foundation for the Enhancement of Human Intelligence, 1900-S Packard Rd., Ann Arbor, MI 48104.

Some Suggested Answers (frequently there are others):

(a) A∪B (b) B∩(C') (c) A∪C∪B
(d) (B-C)∪D (e) A'∩D (f) (A∪D)-(B∪D)'

COMPUTER RECREATIONS

by D. Van Tassel

Syntax Messages

All programmers get tired of getting syntax error messages, but there is an interesting program to write where the goal is to get syntax error messages. Now any fool can get a lot of syntax messages (just forget to declare an array) but try to get the maximum number of *different* syntax error messages. To find out how many error messages are possible check your manuals for a complete listing of syntax error messages.

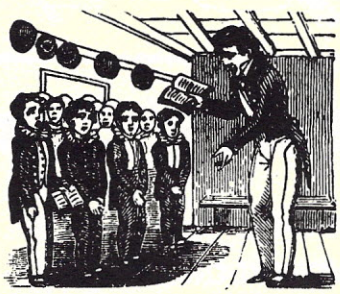
In order to not make it too easy let's try to get as many different syntax messages with as few statements as possible. We can set up a ratio as follows:

$$\text{ratio} = \frac{\text{\# of statements}}{\text{\# of different error messages}}$$

If you get a good solution send me the listing (but you must do the counting). If I get some real good solutions I will publish them in a later column. This problem is language dependent so I will try to publish solutions by language. (Send solutions to D. Van Tassel, Computer Center, Univ. of California, Santa Cruz, CA 95064).

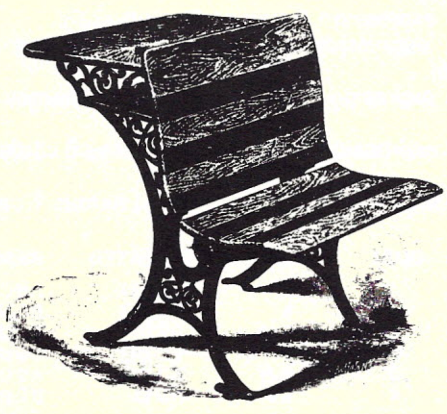
Old Time Problems

Here are several problems from two arithmetic textbooks from around the turn of the century, i.e., around 1800! One is *Schoolmaster's Assistant* by Nathan Daboll, and the other is *An Introduction to Arithmetic* by Erastus Root, 1796. Naturally, there were no computers, calculators, or slide rules in those days, yet consider the fact that these problems do not "come out even" and require a great deal of tedious hand calculation. Try them by hand and then write short programs to solve them. Which do you like best?



1. An ignorant fop wanting to purchase an elegant house, a facetious gentleman told him he had one which he would sell him on these moderate terms, viz. that he should give him a penny for the first door, 2¢ for the second, 4¢ for the third, and so on, doubling at every door, which were 36 in all. It is a bargain, cried the simpleton, and here is a dollar to bind it. Pray, what would the house have cost him? (Can you solve this problem with a 4-line BASIC program?)

2. What is the difference between six dozen dozen and half a dozen dozen?



3. Divide 4½ gallons of brandy equally among 144 soldiers.

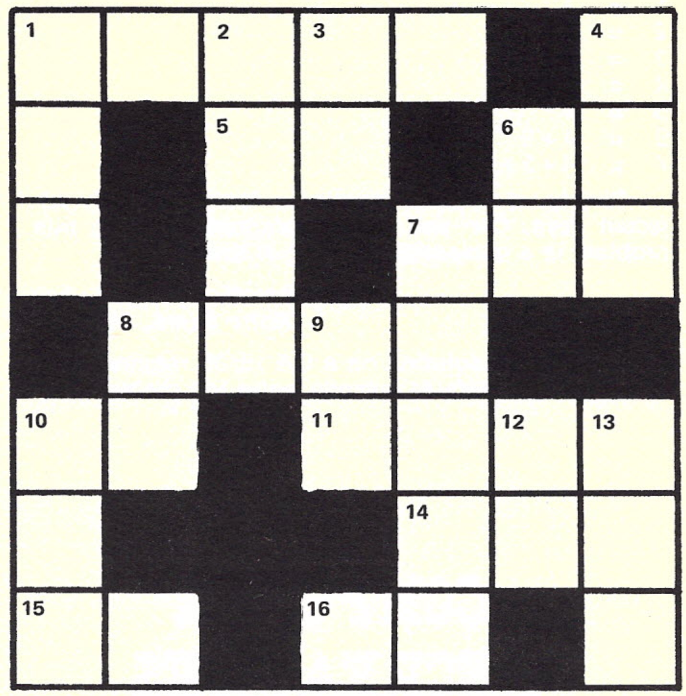
4. How much shalloon that is 3/5 yard wide, will line 5½ yards of camblet which is 1¼ yard wide?

The Little Pigsby Farm Puzzle

The farm known as Little Pigsby has been in the possession of the Dunk Family for several centuries. One of the fields of this farm is rectangular and is known as Dog's Mead. Below are a number of clues to figures relating to the property which must be written in the appropriate places in the framework. When completed "2 down" will give the square of the age of Mrs. Gooby, Farmer Dunk's mother-in-law.

CLUES

The year of the puzzle is 1939.
4840 sq. yds. equal 1 acre.
4 roods equal 1 acre.
20 shillings equal 1 pound sterling.



ACROSS

1. Area of Dog's Mead in square yards.
5. Age of Farmer Dunk's daughter Martha.
6. Difference in yds. between length and breadth of Dog's Mead.
7. No. of roods in Dog's Mead times (X) "8 down."
8. Year when Little Pigsby came into possession of the Dunks.
10. Farmer Dunk's age.
11. Year of birth of Mary, Farmer Dunk's youngest child.
14. Perimeter in yds. of Dog's Mead.
15. Cube of Farmer Dunk's walking speed in M.P.H.
16. "15 across" minus (—) "9 down."

DOWN

1. Value in shillings per acre of Dog's Mead.
3. Age of Mary.
4. Value of Dog's Mead in pounds sterling.
6. Age of Farmer Dunk's first born, Ted, who was twice as old as Mary in 1935.
7. Square of the no. of yds. in breadth of Dog's Mead.
8. No. of minutes it takes Farmer Dunk to walk one and 1/3 times around Dog's Mead.
9. See "10 down."
10. "10 across" times (X) "9 down."
12. One more than the sum of the digits in "10 down."
13. Length of tenure in years of Little Pigsby by the Dunks.

Music More Music

by
J. Quentin Kuyper

Language: HP 2000/Access BASIC

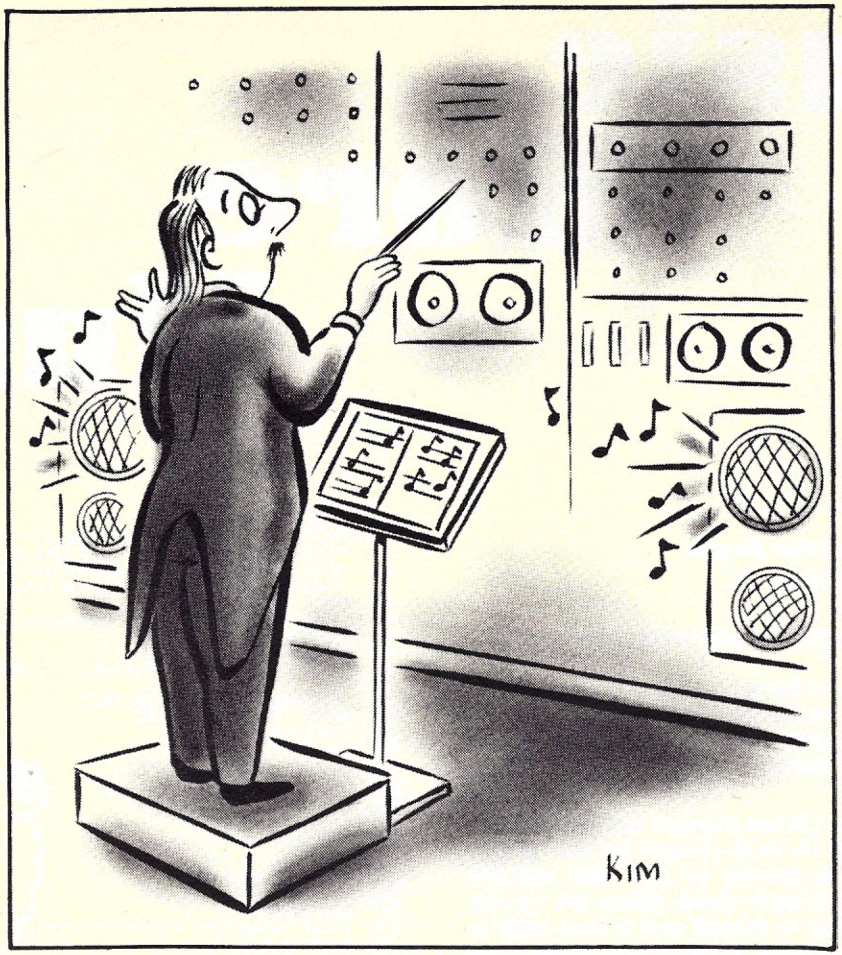
Restrictions:

1) Requires a CRT terminal with cursor addressing capability.

2) 'ESCAPE' sequences for cursor addressing and reverse video are written for use on a SUPERBEE terminal. In order for this program to be usable on other terminals, these sequences must be rewritten to the specifications of those terminals.

USE:This program prints a 5-line musical staff with either treble or bass clef sign. A note chosen by the BASIC language RND function is printed on this staff and the student is asked to type its letter name. When the correct answer is given, the note is erased and a new note is printed. No ledger lines are used. A note is never used twice in a row. A tally is kept of the number of correct responses and this is reported to the student upon completion of 50 items. The student may cause the program to skip to the end at any time by typing the word 'stop'.

```
10  REM          MUSIC READING SKILLS PROGRAM
20  REM          Written by J. Quentin Kuyper
30  REM               Iowa City, Iowa
40  REM              September 10, 1976
50  REM
60  PRINT "This program helps you to practice the spelling of notes in"
70  PRINT "your choice of either treble or bass clef."
80  PRINT LIN(2);"HERE ARE THE DIRECTIONS:"
90  PRINT " 1) When you get a '?', type the letter name of the note pri
     nted on the staff."
100  PRINT " 2) Use either upper or lower case letters."
110  PRINT " 3) If you give up, type '99'. I'll tell you the answer."
120  PRINT " 4) There are 50 questions per session."
130  PRINT " 5) If you want to quit before that, type 'stop'."
140  PRINT LIN(1);"Hit RETURN when you are ready to start."
150  ENTER 255,Z5,Z5
160  PRINT '27"E"
170  DIM Q$[15],P$[15],S$[15],C$[9],T$[13],B$[5]
180  D=J=N3=I=R=0
190  C$="-\!/-\!/."
200  P$='27"F040013"
210  Q$='27"F000013"
220  PRINT
230  PRINT SPA(23);"----------------------------";'10
240  D=D+1
250  IF D<5 THEN 230
260  PRINT LIN(2);"Which clef would you prefer to start with?"
270  PRINT "  1) Treble clef"
280  PRINT "  2) Bass clef"
290  PRINT "  3) It doesn't matter to me."
300  PRINT "(Please type the number of your choice)";
310  ENTER 200,Z5,T
320  GOTO T OF 330,380,420
330  PRINT '27"F026012";
340  PRINT  USING "#,"'34"\_/"'34",8("'34'8'27"A!"'34"),"'34'8'27"A\"'8
     '8'27"A<"'27"A/\"'10">"'8'8'10"/"'8'34
350  PRINT  USING "#,3("'34'8'8'10"/"'34"),"'34'8'8'10"("'8'10"("'10"\"
     '10"\_!_/"'27"A;"'34
360  PRINT  USING '34'27"A"'8"!"'27"A"'8"."'8'8"-"'8'8"-"'8'8'8"-"'8'8"
     ,"'10'8"!"'10"\_/"'34
370  GOTO 410
380  PRINT '27"F025009";
390  PRINT  USING "#,"'34"\_/"'34",2("'34'27"A/"'34"),"'34'27"C"'34",3(
     "'34'8'27"A)"'34"),"'34'8'8'27"A\"'34
400  PRINT  USING "#,3("'34'8'8"'"'34"),"'34'8'8"/"'8'8'10"("'10"\_/"'2
     7"A)"'27"A"'27"C"'27"C*"'8'10'10"*"'34
410  GOTO T OF 730,710
420  PRINT Q$;"OK. I'll flip the magic coin.";'27"J"
430  ENTER 1,Z5,Z5
440  READ N1,N2
450  DATA 1,21,1,20,2,19,3,18,5,17,7,16,9,15,13,14,15,14,19,15,21,16,23
     ,17
460  DATA 24,18,24,19,22,20,22,21
470  PRINT  USING 480;N1,N2
480  IMAGE#,",","F",3d,3d
490  J=J+1
500  IF J=15 THEN 570
510  IF J=16 THEN 600
520  FOR T=1 TO 9
530  PRINT C$[T,T];'27"D";
540  NEXT T
550  PRINT
560  GOTO 440
570  PRINT C$[1,5]
580  ENTER 1,Z5,Z5
590  GOTO 440
600  PRINT "plop"'7'7'7'7'7'7'7'7'7'7;
610  T=INT(RND(1)*2)+1
620  ENTER 1,Z5,Z5
630  IF T=2 THEN 660
640  S$=" HEADS TREBLE "
650  GOTO 670
660  S$=" TAILS BASS "
670  PRINT "   Its";S$[1,6];"! We'll do";S$[7];"clef this time.";'27"K"
680  ENTER 3,Z5,Z5
690  PRINT Q$;'27"K"
700  GOTO 320
710  T$="BAGFEDCBAGF"
720  GOTO 740
730  T$="GFEDCBAGFED"
740  C$="O"
750  I=I+1
760  N5=N4=0
770  IF I=51 THEN 1090
780  N2=11-(INT(RND(1)*11))
790  IF N2=N3 THEN 780
800  N3=N2
810  PRINT  USING 820;40,N2,C$
820  IMAGE #,",","F",3d,3d,a
830  IF N5=1 THEN 740
840  PRINT P$;'27"J";
850  INPUT B$
860  B$=UPS$(B$)
870  IF B$[1,2]="ST" THEN 1090
880  IF B$#"99" THEN 940
890  N4=1
900  PRINT Q$;'27"J";P$;"That one is ";T$[N2,N2]
910  PRINT '10;SPA(20);"Hit RETURN when you are ready to continue.";
920  ENTER 255,Z5,Z5
930  GOTO 1030
940  IF B$=T$[N2,N2] THEN 990
950  PRINT SPA(40);"Wrong. Try again.  -->";
960  N4=N4+1
970  IF N4>8 THEN 900
980  GOTO 850
990  PRINT P$;'27"_2 Correct!! "'27"_3";'27"J"
1000  ENTER 1,Z5,Z5
1010  IF N4#0 THEN 1030
1020  R=R+1
1030  IF N2/2=INT(N2/2) THEN 1060
1040  C$=" "
1050  GOTO 1070
1060  C$="-"
1070  N5=1
1080  GOTO 810
1090  IF N4=0 THEN 1110
1100  GOTO 1120
1110  I=I-1
1120  PRINT Q$;'27"J";"You got";R;"out of";I;"items correct."
1130  IF I=50 AND R/I=1 THEN 1160
1140  PRINT "Practice makes perfect!"
1150  GOTO 1170
1160  PRINT '27"_2 Very Good! "'27"_3"
1170  END
```



A Musical Number Guessing Game

Kurt Inman

This is a number guessing game for the Altair 8800 (or other 8080 based machine). When you guess the right number, the Altair plays the song "DAISY" over the radio. It requires no connections between the computer and the radio. This program is written in machine language for the Altair 8800 with 256 words of memory.

OPERATING INSTRUCTIONS

1. Deposit the number to be guessed in address 002.
2. Deposit your guess in address 1000.
3. Place radio on top of the Altair and tune it to 550-700 KC and turn it on.
4. Hit 'RUN'. If computer plays "DAISY," then you have guessed the number. Rotate the radio to find best reception. If INTE light comes on, hit STOP and examine address 047. If all lights go on then your guess was too big. If none of the lights go on then your guess was too small. Go back to step 2 until you guess the right number.

Kurt Inman can be found at 350 Nelson Road, Scotts Valley, CA 95066. This program originally appeared in PCC (Vol. 5, No. 1), P.O. Box 310, Menlo Park, CA 94025.

Musical Number Guessing Game

Code for 8080 Systems.

ADDRESS	MNEMONIC	CODE	EXPLANATION
000	LXI H	041	Load number to be guessed into register H
001	b2	000	
002	b3	( )	
003	LDA	072	Load guess into accumulator from address 1000
004	b2	000	
005	b3	001	
006	MOV B,A	107	Move accumulator to register B
007	SUB H	224	Subtract register H from accum.
010	JZ	312	Jump if above result = 0 to address 030
011	b2	030	
012	b3	000	
013	JP	362	Jump to address 023 if result is greater than zero.
014	b2	023	
015	b3	000	
016	MVI A	076	Move the data in next line into the accumulator
017	b2	000	
020	JMP	303	Jump to address 036
021	b2	036	
022	b3	000	
023	MVI A	076	Move the data in the next line into the accumulator
024	b2	377	
025	JMP	303	Jump to address 036
026	b2	036	
027	b3	000	
030	CALL	315	Go to 'DAISY' subroutine
031	b2	050	
032	b3	000	
033	JMP	303	Jump to address 041
034	b2	041	
035	b3	000	
036	STA	062	Store accumulator (answer) in 047
037	b2	047	
040	b3	000	
041	EI	373	Turn INTE light on
042	JMP	303	Jump to address 041
043	b2	041	
044	b3	000	
050	LXI H	041	Load address of first music data entry
051	b2	110	
052	b3	000	
053	MOV A,M	176	The following lines (053-247) make the music
054	CPI	376	
055	b2	377	
056	JZ	312	
057	b2	050	
060	b3	000	
061	MVI D	026	
062	b2	040	
063	DCR B	005	
064	JNZ	302	
065	b2	070	
066	b3	000	
067	MOV B,M	106	
070	DCR C	015	
071	JNZ	302	
072	b2	063	
073	b3	000	
074	DCR D	025	
075	JNZ	302	
076	b2	063	
077	b3	000	
100	INR L	054	
101	JMP	303	
102	b2	053	
103	b3	000	

Data for "Daisy"

Address	Data	Address	Data
110	034	170	040
111	034	171	042
112	034	172	046
113	042	173	034
114	042	174	034
115	042	175	042
116	053	176	046
117	053	177	053
120	053	200	053
121	071	201	053
122	071	202	053
123	071	203	046
124	063	204	042
125	055	205	042
126	053	206	053
127	063	207	063
130	063	210	063
131	053	211	053
132	071	212	063
133	071	213	071
134	071	214	071
135	071	215	071
136	071	216	071
137	071	217	071
140	046	220	053
141	046	221	053
142	046	222	042
143	034	223	046
144	034	224	046
145	034	225	071
146	042	226	053
147	042	227	053
150	042	230	042
151	053	231	046
152	053	232	042
153	053	233	040
154	063	234	034
155	055	235	042
156	053	236	053
157	046	237	046
160	046	240	046
161	042	241	071
162	046	242	053
163	046	243	053
164	046	244	053
165	046	245	053
166	046	246	002
167	042	247	377

Scales

All you've ever wanted to see (if its 11) and didn't have a program to ask for...

by
Marvin S. Thostenson

Language: HP 2000/Access BASIC

To practice spelling and observe the differences among the scales, use this program. It generates 11 types of scales: major, natural minor, harmonic minor, Hungarian minor, dorian, phygian, lydian, mixolydian, locrian, and whole tone.

When you run this program, you will be asked, "Which type of scale is wanted?" Respond by typing the first two letters of the name of the desired scale followed immediately by the desired key. Use a lower case (b' for the flat and use '#' for the sharp. Sample in puts would be *phe* for phrygian starting on E, *maf#* for major on F-sharp, and *whg* for whole tone on G.

The Author, Marvin S. Thostenson, is at the School of Music, University of Iowa, Iowa City, Iowa S2242

Sample Run

```
ELEVEN SCALE TYPES -- MAJOR, MINOR, MODAL, AND WHOLE TONE

This program prints in letter names, one octave upward, the major,
the natural, harmonic, melodic, and Hungarian minors, the
dorian, phrygian, lydian, mixolydian, and locrian modes, and
the whole tone scales.

Use a 3- or 4-character input; the first 2 char's are the scale
type, and the 3rd char'r is the single letter tonic, or the
last two char's are the tonic degree or the key signature.
SCALE TYPES-- ma na ha me do ph ly mi lo hu and wh
Input either a tonic or a signature.
EXAMPLES: macb lydb mieb whgb naf# hag# mea# loc# doc phd hue

WHICH TYPE OF SCALE IS WANTED?
?madb

   SCALE ASKED --------Major scale on Db

   ANSWER (in letter names) ----------

        Db  Eb  F   Gb  Ab  Bb  C   Db

WHICH TYPE OF SCALE IS WANTED?
?nae

   SCALE ASKED --------Nat'l minor scale on E

   ANSWER (in letter names) ----------

        E   F#  G   A   B   C   D   E

WHICH TYPE OF SCALE IS WANTED?
?ha2#

   SCALE ASKED --------Harm'c minor scale on B

   ANSWER (in letter names) ----------

        B   C#  D   E   F#  G   A#  B

WHICH TYPE OF SCALE IS WANTED?
?doeb

   SCALE ASKED --------Dorian mode on Eb

   ANSWER (in letter names) ----------

        Eb  F   Gb  Ab  Bb  C   Db  Eb
```

Program Listing

```
10  REM *+* GENERATOR PROGRAM FOR ELEVEN TYPES OF SCALES ***
20  REM Written by Marvin S. Thostenson, Assoc. Prof., School
30  REM of Music, University of Iowa, Iowa City, Iowa 52242.
60  DIM A$[10],B$[50],C$[50],D$[65],E$[65]
70  DIM F$[72],G$[72],H$[72],I$[60],J$[50]
130  DIM K$[40],L$[40],M$[50],N$[40],O$[40]
133  DIM B1$[20],C1$[10],D1$[10],E1$[10],F1$[10]
134  DIM G1$[10],H1$[10],I1$[10],AO$[30],BO$[20],OO$[30]
140  A=B=C=D=E=H=K=L=M=N=O=P=Q=R=T=U=V=X=Y=Z=0
200  W=4
210  PRINT "ELEVEN SCALE TYPES -- MAJOR, MINOR, MODAL, AND WHOLE TONE";LIN(1)
220  PRINT "This program prints in letter names, one octave upward, the major,"
225  PRINT "the natural, harmonic, melodic, and Hungarian minors, the"
227  PRINT "dorian, phrygian, lydian, mixolydian, and locrian modes, and"
229  PRINT "the whole tone scales.";LIN(1)
230  PRINT "Use a 3- or 4-character input; the first 2 char's are the scale"
232  PRINT "type, and the 3rd char'r is the single letter tonic, or the"
234  PRINT "last two char's are the tonic degree or the key signature."
240  PRINT "SCALE TYPES-- ma na ha me do ph ly mi lo hu and wh"
250  PRINT "Input either a tonic or a signature."
260  PRINT "EXAMPLES: macb lydb mieb whgb naf# hag# mea# loc# doc phd hue"
270  B$="SCALE ASKED --------"
280  C$="ANSWER (in letter names) ----------"
290  O$[1,14]="STRUCTURE---- "
300  K$=" tetrachords"
310  U=1
320  PRINT TAB(8);LIN(2),"WHICH TYPE OF SCALE IS WANTED?"
330  INPUT A$
340  N=LEN(A$)
350  IF A$="stop" THEN 1290
360  E$="manahamedophlymilohuwh"
370  FOR X=1 TO 22 STEP 2
380  IF A$[1,2]=E$[X,X+1] THEN 400
390  NEXT X
400  Q=(X+1)/2
410  AO$=A$
420  AO$[1,1]=UPS$(AO$[1,1])
430  READ D$
440  AO$=A$
450  AO$[1,1]=UPS$(AO$[1,1])
460  IF D$[1,2]=AO$[1,2] THEN 480
470  GOTO 430
480  J$=D$
490  RESTORE
500  IF N=3 THEN 520
510  GOTO 530
520  A$[4,4]=" "
530  IF Q=1 OR Q=7 OR Q=8 OR Q=11 THEN 550
540  IF Q >= 2 AND Q <= 6 OR Q=9 OR Q=10 THEN 570
550  Y=1
560  GOTO 580
570  Y=2
580  F$="bxexaxdxgxcxfxb#e#a#d#g#c#f#b e a d g c f bbebabdbgbcbfbbdedadddgdcd"
590  G$="BxExAxDxGxCxFxB#E#A#D#G#C#F#B E A D G C F BbEbAbDbGbCbFbBdEdAdDdGdCd"
600  GOTO Y OF 610,630
610  H$="5t4t3t2t1t7x6x5x4x3x2x1x7#6#5#4#3#2#1#0#1b2b3b4b5b6b7b1d2d3d4d5d6d7d"
620  GOTO 640
630  H$="2t1t7x6x5x4x3x2x1x7#6#5#4#3#2#1#0#1b2b3b4b5b6b7b1d2d3d4d5d6d7d8d9d  "
640  FOR V=1 TO 68 STEP 2
650  IF A$[3,4]=F$[V,V+1] THEN 680
660  IF A$[3,4]=H$[V,V+1] THEN 680
670  NEXT V
680  C1$=G$[V,V+1]
690  T=T+1
700  IF T=9 THEN 1160
710  GOTO T OF 720,740,790,840,890,940,990,1040
720  R=0
730  GOTO 1060
740  IF Q=6 OR Q=9 THEN 770
750  R=-4
760  GOTO 1060
770  R=10
780  GOTO 1060
790  IF Q=1 OR Q=7 OR Q=8 OR Q=11 THEN 820
800  R=6
810  GOTO 1060
820  R=-8
830  GOTO 1060
840  IF Q=7 OR Q=10 OR Q=11 THEN 870
850  R=2
860  GOTO 1060
870  R=-12
880  GOTO 1060
890  IF Q=9 'OR Q=11 THEN 920
900  R=-2
910  GOTO 1060
920  R=12
930  GOTO 1060
940  IF Q=1 OR Q=4 OR Q=5 OR Q=7 OR Q=8 THEN 970
950  R=8
960  GOTO 1060
970  R=-6
980  GOTO 1060
990  IF Q=1 OR Q=3 OR Q=4 OR Q=7 OR Q=10 THEN 1020
1000  R=4
1010  GOTO 1060
1020  R=-10
1030  GOTO 1060
1040  R=0
1050  GOTO 1060
1060  IF Q=11 AND T=5 THEN 1090
1070  I$[U,U+1]=G$[V+R,V+R+1]
1080  GOTO 1100
1090  GOTO 690
1100  I$[U+2,U+3]="  "
1110  IF I$[U+1,U+1]="d" THEN 1130
1120  GOTO 1140
1130  I$[U+1,U+2]="bb"
1140  U=U+4
1150  GOTO 690
1160  PRINT LIN(1),TAB(3),B$;J$;C1$;LIN(1)
1170  PRINT TAB(3);C$;LIN(1)
1180  PRINT LIN(1);TAB(8);I$
1190  C1$=""
1200  I$=""
1210  G$=""
1220  Q=T=R=0
1230  PRINT
1240  GOTO 310
1250  DATA "Major scale on ","Nat'l minor scale on ","Harm'c minor scale on "
1260  DATA "Mel'c minor scale on ","Dorian mode on ","Phrygian mode on "
1270  DATA "Lydian mode on ","Mixolydian mode on ","Locrian mode on "
1280  DATA "Hung'n minor scale on ","Whole tone scale on "
1290  END
```

Sharps & Flats

by
J. Quentin Kuyper

Language: HP 2000/Access BASIC

Restrictions: None; usable on all CRT's and TTY's.

Use: Practice in naming key signatures. The computer names a key (such as F# minor — chosen by the BASIC language RND function) and the student is asked to respond with the correct key signature for that key. Since a knowledge of key signatures, in order to be useful, must be almost immediate, the program keeps track of how much time the student requires to complete 20 items. This datum, along with the number of correctly answered items, is reported to the student upon completion of the program.

Program Listing

```
10  REM   KEY SIGNATURE DRILL PROGRAM
20  REM
30  REM   Written by J. Quentin Kuyper
40  REM         Iowa City, Iowa
50  REM        September 10, 1976
60  REM
70  PRINT LIN(2);"This session is designed to help you improve your kn
     owledge"
80  PRINT "of KEY SIGnatures. Good luck!";LIN(2)
90  PRINT "HERE ARE THE DIRECTIONS:";LIN(1)
100  PRINT " 1) I will print a key, such as Db minor."
110  PRINT " 2) You will respond by typing one number and a symbol"
120  PRINT "    representing the correct key signature."
130  PRINT " 3) Use '#' for a sharp sign and 'b' (lower case B) for a
     flat sign."
140  PRINT "     (For example, some answers might be 3b, 5#, or 0b.)"
150  PRINT " 4) If you give up on a question, type '99' and I'll tell
     you the answer."
160  PRINT " 5) We normally do 20 items and then stop. If you want to
     quit"
170  PRINT "    before you have done all 20, type the word 'stop'. "
180  PRINT LIN(1);"Hit the RETURN button when you are ready to start."
     ;LIN(1)
190  ENTER 255,Z5,Z5
200  T1=T2=T3=T4=T5=T6=M1=H1=0
210  T4=TIM(4)
220  T2=TIM(0)*60
230  T1=TIM(1)*3600
240  IF T6=1 THEN 280
250  T5=T4+T2+T1
260  T6=1
270  GOTO 360
280  IF (T1+T2+T4) >= T5 THEN 300
290  T1=T1+86400.
300  T3=(T4+T2+T1)-T5
310  H1=INT(T3/3600)
320  T3=T3-(H1*3600)
330  M1=INT(T3/60)
340  T3=T3-(M1*60)
350  GOTO 1160
360  DIM M[18,2],T$[7],C$[36],A$[5],D$[9]
370  I=H=B=C=V1=0
380  B=8
390  S1=0
400  W=0
410  H=H+1
420  IF H>2 THEN 550
430  W=W+1
440  IF W>18 THEN 530
450  IF S1=1 THEN 480
460  B=B-1
470  GOTO 490
480  B=B+1
490  M[W,H]=B
500  IF B#0 THEN 430
510  S1=1
520  GOTO 430
530  B=B+1
540  GOTO 390
550  K=K2=K4=K5=0
560  D$="123456789"
570  C$="CbGbDbAbEbBbF C G D A E B F#C#G#D#A#"
580  T$=" Major?"
590  K=INT(RND(1)*2)+1
600  I=I+1
610  PRINT I;
620  K3=K4=K6=0
630  K2=INT(RND(1)*15)
640  IF K2=V1 THEN 630
650  IF K2#0 THEN 670
660  K2=15
670  V1=K2
680  IF K=1 THEN 710
690  K2=K2+3
700  T$=" Minor?"
710  K5=5*((I/5)-INT(I/5))
720  GOTO K5+1 OF 810,730,750,770,790
730  PRINT "What is the key signature of ";
740  GOTO 820
750  PRINT "How about ";
760  GOTO 820
770  PRINT "Try ";
780  GOTO 820
790  PRINT "The next one is ";
800  GOTO 820
810  PRINT "And now ";
820  PRINT C$[2*K2-1,2*K2];T$
830  INPUT A$
840  IF K6=1 THEN 910
850  IF K=1 THEN 870
860  K3=3
870  IF K2-K3<9 THEN 900
880  B$="#"
890  GOTO 910
900  B$="b"
910  IF A$="stop" THEN 1030
920  IF A$="99" THEN 1040
930  IF M[K2,K]#0 THEN 960
940  IF A$[1,1]="0" THEN 980
950  GOTO 1130
960  IF A$[1,1]#D$[M[K2,K],M[K2,K]] THEN 1130
970  IF A$[2,2]#B$ THEN 1130
980  PRINT "Correct!"
990  IF K6=1 THEN 1010
1000  C=C+1
1010  IF I<20 THEN 580
1020  GOTO 210
1030  K4=1
1040  IF M[K2,K]#0 THEN 1070
1050  PRINT "You should know that one. It is 0b and 0#."
1060  GOTO 1080
1070  PRINT "The correct answer to that one is "D$[M[K2,K],M[K2,K]];B$
1080  IF K4=0 THEN 580
1090  IF K6=1 THEN 1110
1100  I=I-1
1110  GOTO 210
1120  GOTO 1010
1130  PRINT "Wrong! Try again or type '99'"
1140  K6=1
1150  GOTO 830
1160  PRINT "You got";C;"out of the ";I;"items you tried correct on th
     e first try.";LIN(2)
1170  PRINT "It took you";
1180  IF H1=0 THEN 1200
1190  PRINT H1;"hours,";
1200  IF M1=0 THEN 1230
1210  PRINT M1;"minutes,";
1220  PRINT " and";
1230  PRINT T3;"seconds to do";I;"items."
1240  END
```

Musical Magic Squares

by
Fred T. Hofstetter

In music, the term "magic square" refers to the compositional matrix used by composers of 12-tone music. The twelve-tone school was started by Arnold Schoenberg, who discovered that if music was composed such that one of the twelve tones is repeated until every other tone is used, an atonal texture would result. Atonal means that there is no tonic, or that there is no "do re mi."

A twelve-tone row is a set of numbers from 1 through 12 typed in any order with no repetitions. Twelve-tone music uses a row and permutations of the row. The permutations consist of transpositions which are the pattern of the original row started on different notes; retrogrades which are the original row and the transpositions played backwards; inversions which are mirror-images of the original row and its transpositions; and retrograde inversions which are the inversions played backwards. The magic square program asks for a twelve-tone row. It then computes the transpositions, inversions, retrograde inversions, and retrogrades of the row, and prints them out as a matrix. The matrix is read as follows:

1. Reading rows from left to right yields transpositions.
2. Reading rows from right to left yields retrogrades.
3. Reading columns from top to bottom yields inversions.
4. Reading columns from botton to top yields retrograde inversions.

Two special symbols are used in the matrix: N means natural and ? means flat. It takes about half an hour to make a magic square by hand. The computer does it in just a second.

Program Listing

```
PRINT MASQUA

1000 REM    MAGIC SQUARE GENERATOR
1010 REM
1020 DEF FNM(X,Y)=X-Y*INT(X/Y)
1030 REM    MODULUS FUNCTION GIVES THE REMAINDER OF X DIVIDED BY Y
1040 DIM A(12)
1050 PRINT
1060 PRINT
1070 PRINT "WHAT ARE THE ROW NUMBERS";
1080 MAT INPUT A
1090 REM    INPUT THE MAIN ROW
1100 PRINT
1110 PRINT
1120 PRINT
1130 PRINT "             M A G I C   S Q U A R E"
1140 PRINT
1150 PRINT
1160 LET B=1
1170 REM    B IS THE COUNTER FOR THE ROWS OF THE SQUARE
1180 LET C=0
1190 REM    C KEEPS TRACK OF THE DIFFERENCES BETWEEN SUCCESSIVE ROWS
1200 FOR D=1 TO 12
1210 REM    D IS THE COUNTER FOR INDIVIDUAL ELEMENTS OF THE ROWS
1220 LET E=FNM(C+A(D)-1,12)
1230 REM    E IS THE ACTUAL NOTE THAT SHOULD BE PRINTED
1240 IF E=0 THEN PRINT "CN  ";
1250 IF E=1 THEN PRINT "C#  ";
1260 IF E=2 THEN PRINT "DN  ";
1270 IF E=3 THEN PRINT "E?  ";
1280 IF E=4 THEN PRINT "EN  ";
1290 IF E=5 THEN PRINT "FN  ";
1300 IF E=6 THEN PRINT "F#  ";
1310 IF E=7 THEN PRINT "GN  ";
1320 IF E=8 THEN PRINT "A?  ";
1330 IF E=9 THEN PRINT "AN  ";
1340 IF E=10 THEN PRINT "B?  ";
1350 IF E=11 THEN PRINT "BN  ";
1360 NEXT D
1370 PRINT
1380 PRINT
1390 IF B>11 GOTO 1050
1400 REM    IF FINISHED, ASK FOR THE NEXT SQUARE. . .
1410 LET B=B+1
1420 LET C=C+A(B-1)-A(B)+12
1430 REM    OTHERWISE, GO BACK AND DO THE NEXT ROW
1440 GOTO 1200
#
```

N = Natural
? = Flat
= Sharp

Sample Run

```
WHAT ARE THE ROW NUMBERS?  3,7,11,1,10,8,4,9,6,5,2,12


              M A G I C   S Q U A R E

DN  F#  B?  CN  AN  GN  E?  A?  FN  EN  C#  BN

B?  DN  F#  A?  FN  E?  BN  EN  C#  CN  AN  GN

F#  B?  DN  EN  C#  BN  GN  CN  AN  A?  FN  E?

EN  A?  CN  DN  BN  AN  FN  B?  GN  F#  E?  C#

GN  BN  E?  FN  DN  CN  A?  C#  B?  AN  F#  EN

AN  C#  FN  GN  EN  DN  B?  E?  CN  BN  A?  F#

C#  FN  AN  BN  A?  F#  DN  GN  EN  E?  CN  B?

A?  CN  EN  F#  E?  C#  AN  DN  BN  B?  GN  FN

BN  E?  GN  AN  F#  EN  CN  FN  DN  C#  B?  A?

CN  EN  A?  B?  GN  FN  C#  F#  E?  DN  BN  AN

E?  GN  BN  C#  B?  A?  EN  AN  F#  FN  DN  CN

FN  AN  C#  E?  CN  B?  F#  BN  A?  GN  EN  DN
```

Another new game from Creative Computing . . . .

DODGEM

by Mac Oglesby

DODGEM is a game originally devised in 1972 by Colin Vout, a student at the University of Cambridge, England. It got its major publicity from Martin Gardner who discussed it in the June 1975 issue of *Scientific American*.

This version of DODGEM is written in Dartmouth BASIC. It may be played by two players in which case the computer is the referee, or by one player against the computer in which case you'd better watch out!

```
1 • • •
2 • • •
3 • • •
• A B C
```

The instructions for DODGEM may be found in Lines 3120-3500 of the program listing.

Sample Run

```
WANT INSTRUCTIONS FOR DODGEM? NO
BOARD SIZE (3-6)? 4
HOW MANY PLAYERS  (1 OR 2)? 1
OK, THE COMPUTER WILL MOVE THE DIGITS.
WHO MOVES FIRST  (1=COMPUTER  2=YOU)? 1

HERE WE GO...

 1 . . .
 2 . . .
 3 . . .
 . A B C

THE DIGITS MOVE:   3E
LETTERS MOVE? H
THE LETTERS HAVE THESE LEGAL MOVES:
  AW  BN  CN
LETTERS MOVE? CN

 1 . . .
 2 . . .
 . 3 . C
 . A B .

THE DIGITS MOVE:   1E
LETTERS MOVE? BN

 . 1 . .
 2 . . .
 . 3 B C
 . A . .

THE DIGITS MOVE:   1E
LETTERS MOVE? CN

 . . 1 .
 2 . . C
 . 3 B .
 . A . .

THE DIGITS MOVE:   2E
LETTERS MOVE? CN

 . . 1 C
 . 2 . .
 . 3 B .
 . A . .

THE DIGITS MOVE:   2E
LETTERS MOVE? BN
ILLEGAL MOVE OR BAD INPUT.
INPUT IGNORED.  TYPE H FOR HELP.
LETTERS MOVE? BE

 . . 1 C
 . . 2 .
 . 3 . B
 . A . .

THE DIGITS MOVE:   2E
LETTERS MOVE? R
THE LETTERS GIVE UP!!
*** THE DIGITS WIN!!!
```

Program Listing

```
100 ' NAME:  ELEMLIB***:DODGEM
110 '
120 ' BY:  MAC OGLESBY ON 18 OCT 75.
130 '
140 ' DESCRIPTION:  TWO SETS OF PIECES RACE AT RIGHT ANGLES ACROSS A
150 '   SQUARE BOARD.  FOR ONE OR TWO PLAYERS.
160 '
170 ' INSTRUCTIONS:  TYPE "RUN" FOR COMPLETE INSTRUCTIONS.
180 '
190 ' REMARKS:  THE GAME OF DODGEM IS DESCRIBED BY MARTIN GARDNER
200 '    IN "SCIENTIFIC AMERICAN" OF JUNE 1975.
210 '
220 '
230 RANDOMIZE
240 PRINT "WANT INSTRUCTIONS FOR DODGEM";
250 LINPUT A$
260 GOSUB 2950                               'PROCESS INPUT
270 IF SEG$(A$,1,1)<>"Y" THEN 290
280    GOSUB 3130                            'INSTRUCTIONS
290 PRINT "BOARD SIZE (3-6)";
300 INPUT A
310 LET A=INT(A)
320 IF (6-A)*(A-3)=>0 THEN 340
330    GOTO 290
340 LET P(1,0)=P(2,0)=A-1                    'HOW MANY PIECES AT START
350
360
370 ' MATRIX P(,) KEEPS TRACK OF PIECES
380 FOR J=1 TO A-1
390    LET P(1,J)=10*J+1                     'LOCATE DIGITS
400 NEXT J
410
420 FOR J=1 TO A-1
430    LET P(2,J)=10*A+J+1                   'LOCATE LETTERS
440 NEXT J
450
460 LET F=1
470 LET M$(1)="NES"                          'LEGAL MOVES FOR THE DIGITS
480 LET M$(2)="NE W"                         'LEGAL MOVES FOR THE LETTERS
490 LET C$(1)="DIGITS"
500 LET C$(2)="LETTERS"
510 LET A$(1)="1234567"
520 LET A$(2)="ABCDEFG"
530
540
550 ' SET UP BOARD
560 FOR J=1 TO A
570    IF J=A THEN 630
580       LET D$(J,1)=CHR$(48+J)
590       FOR K=2 TO A
600          LET D$(J,K)="."
610       NEXT K
620       GOTO 670
630    LET D$(J,1)="."
640    FOR K=2 TO A
650       LET D$(J,K)=CHR$(63+K)
660    NEXT K
670 NEXT J
680
690 PRINT "HOW MANY PLAYERS  (1 OR 2)";
700 INPUT B
710 IF B=2 THEN 800
720    IF B=1 THEN 740
730       GOTO 690
740    PRINT "OK, THE COMPUTER WILL MOVE THE DIGITS."
750    PRINT "WHO MOVES FIRST  (1=COMPUTER  2=YOU)";
760    INPUT F
770    IF (2-F)*(1-F)=0 THEN 800
780       PRINT "PLEASE TYPE 1 OR 2.  NOW, ";
```

```
790       GOTO 750
800 PRINT
810 PRINT "HERE WE GO..."
820
830 ' PRINT THE DISPLAY
840 PRINT
850 FOR J=1 TO A
860    FOR K=1 TO A
870       PRINT " ";D$(J,K);
880    NEXT K
890    PRINT
900 NEXT J
910 PRINT
920
930
940 ' MAIN MOVE LOOP
950 FOR J=F TO 3-F STEP 3-2*F
960
970    ' CHECK IF PLAYER J HAS A LEGAL MOVE LEFT
980    FOR J1=1 TO A-1
990       LET R=INT(P(J,J1)/10)         'ROW OF J1TH PIECE
1000       LET C=P(J,J1)-10*R            'COLUMN OF J1TH PIECE
1010
1020       ON J GOTO 1040,1100
1030
1040       IF C=A THEN 1080              'AT EDGE OF BOARD?
1050          IF C>A THEN 1070           'OFF BOARD?
1060             GOTO 1160               'SEE IF PIECE CAN MOVE
1070          GOTO 1250                  'TRY NEXT PIECE
1080       GOTO 1300                     'LEGAL MOVE EXISTS
1090
1100       IF R=1 THEN 1140              'EDGE OF BOARD?
1110          IF R=0 THEN 1130           'OFF BOARD?
1120             GOTO 1160               'SEE IF PIECE CAN MOVE
1130          GOTO 1250                  'TRY NEXT PIECE
1140       GOTO 1300                     'LEGAL MOVE EXISTS
1150
1160       IF D$(R-1,C)="." THEN 1240  'SEE IF PIECE CAN GO NORTH
1170          IF D$(R,C+1)="." THEN 1240  'SEE IF PIECE CAN GO EAST
1180             IF J=2 THEN 1220
1190                IF D$(R+1,C)="." THEN 1210  'CAN IT GO SOUTH
1200                   GOTO 1250         'TRY NEXT PIECE
1210                GOTO 1300            'LEGAL MOVE EXISTS
1220             IF D$(R,C-1)="." THEN 1240  'SEE IF PIECE CAN GO WEST
1230                GOTO 1250            'TRY NEXT PIECE
1240       GOTO 1300                     'LEGAL MOVE EXISTS
1250    NEXT J1
1260    PRINT "THE ";C$(3-J);" HAVE LEFT NO LEGAL MOVE FOR THE ";C$(J);"!"
1270    PRINT "*** THE ";C$(J);" WIN!!!"
1280    STOP
1290
1300    IF B=2 THEN 1750
1310       IF J=2 THEN 1750
1320
1330          ' GENERATE COMPUTER'S MOVE
1340          LET L1=2
1350          FOR L0=1 TO 3
1360
1370             ON L0 GOTO 1450,1380,1430
1380             IF RND <.5 THEN 1410
1390                LET L1=1
1400                GOTO 1450
1410             LET L1=3
1420             GOTO 1450
1430             LET L1=4-L1
1440
1450             LET P1=INT(RND*A)
1460             FOR L2=1 TO A-1
1470                LET P1=P1+1
1480                IF P1<=A-1 THEN 1500
1490                   LET P1=P1-(A-1)
1500                LET R=INT(P(J,P1)/10)  'ROW
1510                LET C=P(J,P1)-10*R  'COLUMN
1520                IF C>A THEN 1720
1530
1540                   ON L1 GOTO 1570,1620,1690
1550
1560                   ' NORTH
1570                   IF D$(R-1,C)="." THEN 1590
1580                      GOTO 1720
1590                   GOTO 2090
1600
1610                   ' EAST
1620                   IF D$(R,C+1)="." THEN 1660
1630                      IF C=A THEN 1650
1640                         GOTO 1720
1650                      LET P(J,0)=P(J,0)-1
1660                   GOTO 2160
1670
1680                   ' SOUTH
1690                   IF D$(R+1,C)="." THEN 1710
1700                      GOTO 1720
1710                   GOTO 2230
1720             NEXT L2
1730          NEXT L0
1740
1750 PRINT C$(J);" MOVE";
1760 LINPUT A$
1770 GOSUB 2950                          'PROCESS INPUT
1780
1790 IF SEG$(A$,1,1)="R" THEN 2860
1800    IF SEG$(A$,1,1)="H" THEN 2460
1810       LET A$=SEG$(A$,1,2)        'LOOK ONLY AT 1ST 2 CHARACTERS
1820       LET P1=POS(SEG$(A$(J),1,A-1),SEG$(A$,1,1),1)  'WHAT PIECE
1830       IF P1=0 THEN 2430
1840          LET P2=POS(M$(J),SEG$(A$,2,2),1)  'WHAT DIRECTION
1850          IF P2=0 THEN 2430
1860             LET R=INT(P(J,P1)/10)  'ROW OF PIECE
1870             LET C=P(J,P1)-10*R  'COLUMN OF PIECE
1880             IF R=0 THEN 2430
1890                IF C>A THEN 2430
1900
1910                   ON J GOTO 1930,1990
1920
1930                   IF C<A THEN 1970  'PIECE NOT AT EAST EDGE
1940                      IF P2<>2 THEN 1970  'DIRECTION NOT EAST
1950                         LET P(1,0)=P(1,0)-1  'REDUCE # OF PIECES
1960                         GOTO 2160  'UPDATE PIECE LOCATION
1970                   GOTO 2040     'CHECK MOVE FURTHER
1980
1990                   IF R>1 THEN 2040  'PIECE NOT AT NORTH EDGE
2000                      IF P2<>1 THEN 2040  'DIRECTION NOT NORTH
2010                         LET P(2,0)=P(2,0)-1  'REDUCE # OF PIECES
2020                         GOTO 2090  'UPDATE LOCATIONS
2030
2040                   ON P2 GOTO 2070,2140,2210,2280
2050
2060                   ' SEE IF PIECE CAN MOVE NORTH
2070                   IF D$(R-1,C)="." THEN 2090
2080                      GOTO 2430  'ERROR MESSAGE
2090                   LET D$(R-1,C)=SEG$(A$(J),P1,P1)
2100                   LET P(J,P1)=P(J,P1)-10
2110                   GOTO 2330
2120
2130                   ' EAST
2140                   IF D$(R,C+1)="." THEN 2160
2150                      GOTO 2430  'ERROR MESSAGE
2160                   LET D$(R,C+1)=SEG$(A$(J),P1,P1)
```

```
2170                   LET P(J,P1)=P(J,P1)+1
2180                   GOTO 2330
2190
2200                   ' SOUTH
2210                   IF D$(R+1,C)="." THEN 2230
2220                      GOTO 2430  'ERROR MESSAGE
2230                   LET D$(R+1,C)=SEG$(A$(J),P1,P1)
2240                   LET P(J,P1)=P(J,P1)+10
2250                   GOTO 2330
2260
2270                   ' WEST
2280                   IF D$(R,C-1)="." THEN 2300
2290                      GOTO 2430  'ERROR MESSAGE
2300                   LET D$(R,C-1)=SEG$(A$(J),P1,P1)
2310                   LET P(J,P1)=P(J,P1)-1
2320
2330                   LET D$(R,C)="."  'FINISH UPDATING DISPLAY
2340                   IF B=2 THEN 2380
2350                      IF J=2 THEN 2380
2360                         PRINT "THE DIGITS MOVE:  ";SEG$(A$(J),P1,P
2370                         'SEG$(A$(J),P1,P1);SEG$(M$(J),L1,L1)
2380                   IF P(J,0)<>0 THEN 2420  'CHECK FOR WIN
2390                      PRINT
2400                      PRINT "*** THE ";C$(J);" WIN!!!"
2410                      STOP
2420                   GOTO 2900     'GET NEXT PLAYER'S MOVE
2430       PRINT "ILLEGAL MOVE OR BAD INPUT."
2440       PRINT "INPUT IGNORED.  TYPE H FOR HELP."
2450       GOTO 1750
2460    PRINT "THE ";C$(J);" HAVE THESE LEGAL MOVES:"
2470    ' HELP! -- PRINT LEGAL MOVES
2480    FOR J3=1 TO A-1
2490       LET P$=SEG$(A$(J),J3,J3)  'NAME OF PIECE
2500       LET R=INT(P(J,J3)/10)     'ROW
2510       LET C=P(J,J3)-10*R        'COLUMN
2520
2530       ON J GOTO 2550,2620
2540
2550       IF C=A THEN 2590          'PIECE AT EAST EDGE
2560          IF C>A THEN 2580       'OFF BOARD
2570             GOTO 2690           'CHECK FURTHER
2580          GOTO 2830              'TRY NEXT PIECE
2590       PRINT "  ";P$;"E";
2600       GOTO 2690
2610
2620       IF R=1 THEN 2660          'NORTH EDGE
2630          IF R=0 THEN 2650       'OFF BOARD
2640             GOTO 2690
2650          GOTO 2830
2660       GOTO 2700
2670
2680       ' NORTH
2690       IF D$(R-1,C)<>"." THEN 2730
2700          PRINT "  ";P$;"N";
2710
2720          ' EAST
2730       IF D$(R,C+1)<>"." THEN 2770
2740          PRINT "  ";P$;"E";
2750
2760          ' SOUTH AND WEST
2770       IF J=2 THEN 2810
2780          IF D$(R+1,C)<>"." THEN 2800
2790             PRINT "  ";P$;"S";
2800          GOTO 2830
2810       IF D$(R,C-1)<>"." THEN 2830
2820          PRINT "  ";P$;"W";
2830    NEXT J3
2840    PRINT
2850    GOTO 1750                    'TRY AGAIN
2860    PRINT "THE ";C$(J);" GIVE UP!!"
2870    PRINT "*** THE ";C$(3-J);" WIN!!!"
2880    STOP
2890
2900 NEXT J
2910 GOTO 840                        'PRINT DISPLAY
2920
2930
2940 ' PROCESS INPUT; CHANGE LOWERCASE TO UPPERCASE, IGNORE COMMAS, SPACES
2950 IF LEN(A$)>10 THEN 3090
2960    LET C1=0                     'RESET COUNTER
2970    CHANGE A$ TO A
2980    FOR J2=1 TO A(0)
2990       IF A(J2)<96 THEN 3010     'CHECK FOR UPPERCASE
3000          LET A(J2)=A(J2)-32     'CHANGE TO UPPERCASE
3010       IF (57-A(J2))*(A(J2)-48)=>0 THEN 3040  'CHECK FOR A DIGIT
3020          IF (90-A(J2))*(A(J2)-65)=>0 THEN 3040  'CHECK FOR A LETTER
3030             GOTO 3060           'ELSE IGNORE THIS CHARACTER
3040       LET C1=C1+1
3050       LET A(C1)=A(J2)           'STORE THIS CHARACTER
3060    NEXT J2
3070    LET A(0)=C1
3080    CHANGE A TO A$
3090 RETURN
3100
3110
3120 ' INSTRUCTIONS
3130 PRINT
3140 PRINT "HERE'S A SAMPLE PLAYING BOARD:"
3150 PRINT
3160 PRINT "1 . . . ."
3170 PRINT "2 . . . ."
3180 PRINT "3 . . . ."
3190 PRINT "4 . . . ."
3200 PRINT ". A B C D"
3210 PRINT
3220 PRINT "TWO SETS OF PIECES (DIGITS AND LETTERS) RACE AT RIGHT ANGLES"
3230 PRINT "ACROSS A SQUARE BOARD.  VACANT LOCATIONS ARE SHOWN AS PERIODS."
3240 PRINT "YOU GET TO CHOOSE THE BOARD SIZE.  (THE ONE ABOVE IS SIZE 5.)"
3250 PRINT
3260 PRINT "  N"
3270 PRINT "  :"
3280 PRINT "W---E"
3290 PRINT "  :"
3300 PRINT "  S"
3310 PRINT
3320 PRINT "THE OBJECT IS TO MOVE ALL OF YOUR PIECES ACROSS THE BOARD"
3330 PRINT "AND OFF THE OPPOSITE EDGE.  DIGITS LEAVE THE BOARD ONLY AT"
3340 PRINT "THE EASTERN EDGE; LETTERS ONLY AT THE NORTHERN.  THE WINNER"
3350 PRINT "IS THE PLAYER WHOSE PIECES HAVE ALL LEFT THE BOARD."
3360 PRINT
3370 PRINT "THE PLAYERS GO IN TURN, MOVING ONE OF THEIR PIECES TO AN"
3380 PRINT "ADJACENT LOCATION WHICH IS EITHER OFF THE BOARD OR CURRENTLY"
3390 PRINT "VACANT.  THERE ARE NO DIAGONAL MOVES, NO JUMPS AND NO CAPTURES"
3400 PRINT "DIGITS CANNOT MOVE WEST, NOR LETTERS MOVE SOUTH."
3410 PRINT
3420 PRINT "TO MOVE A PIECE, TYPE ITS NAME AND THE FIRST LETTER OF THE"
3430 PRINT "DESIRED DIRECTION.  EXAMPLES:"
3440 PRINT "  2E  MEANS PIECE  2  WANTS TO GO EAST"
3450 PRINT "  BW  MEANS PIECE  B  WANTS TO GO WEST."
3460 PRINT
3470 PRINT "NOTE:  YOU FORFEIT THE GAME IF YOUR MOVE LEAVES YOUR OPPONENT"
3480 PRINT "WITHOUT ANY LEGAL MOVE."
3490 PRINT
3500 PRINT "LASTLY, YOU MAY TYPE R TO RESIGN OR H FOR HELP."
3510 PRINT
3520 RETURN
3530 END
```

* Yes, astute readers — Creative Computing's erudite, effervescent, easy-going Edu-Man is indeed the twin brother of fearless, fighting, foulmouthed Wonder Wart Hog. Apologies to Gilbert Shelton. — DHA

So you've got a personal computer. Now what?

101 Basic Computer Games

David H. Ahl. An anthology of games and simulations—from Acey-Deucey to Yahtzee, all in the BASIC language. Contains a complete listing, sample run, plus a descriptive write-up of each game. Our most popular book! Large format, 248 pp. $7.50 [6C]

What to Do After You Hit Return

Another collection of games and simulations—all in BASIC—including number guessing games, word games, hide-and-seek games, pattern games, board games, business and social science simulations and science fiction games. Large format. 158 pp. $6.95 [8A]

Fun & Games with the Computer

Ted Sage. "This book is designed as a text for a one-semester course in computer programming using the BASIC language. The programs used as illustrations and exercises are games rather than mathematical algorithms, in order to make the book appealing and accessible to more students. The text is well written, with many excellent sample programs. Highly recommended." —*The Mathematics Teacher* 351 pp. $6.95 [8B]

Game Playing With the Computer, 2nd Ed.

Donald Spencer. Over 70 games, puzzles, and mathematical recreations for the computer. Over 25 games in BASIC and FORTRAN are included complete with descriptions, flowcharts, and output. Also includes a fascinating account of the history of game-playing machines, right up to today's computer war games. Lots of "how-to" information for applying mathematical concepts to writing your own games. 320 pp. 1976 $14.95 [8S]

BYTE Magazine

If you are considering a personal computing system now or later, BYTE provides a wealth of information on how to get started at an affordable price. Covers theory of computers, practical applications, and of course, lots of how-to build it. Monthly. 1-Year sub'n $12.00 [2A], 3-Years $30.00 [2B]

Games & Puzzles Magazine

The only magazine in the world devoted to games and puzzles of every kind — mathematical, problematical, crosswords, chess, gomoko, checkers, backgammon, wargames, card games, board games, reviews, competitions, and more. Monthly. 1-Year sub'n $12.00[3A]

Games With The Pocket Calculator

Sivasailam Thiagarajan and Harold Stolovitch. A big step beyond tricks and puzzles with the hand calculator, the two dozen games of chance and strategy in this clever new book involve two or more players in conflict and competition. A single inexpensive four-banger is all you need to play. Large format. 50 pp. $2.00 [8H]

Games, Tricks and Puzzles For A Hand Calculator

Wally Judd. This book is a necessity for anyone who owns or intends to buy a hand calculator, from the most sophisticated (the HP65, for example) to the basic "four banger." 110 pp. $2.95 [8D]

Creative Computing Magazine

So you've got your own computer. Now what? *Creative Computing* is chock full of answers — new computer games with complete listings every issue, TV color graphics, simulations, educational programs, how to catalog your LPs on computer, etc. Also computer stories by Asimov, Pohl, and others; loads of challenging problems and puzzles; in-depth equipment reports on kits, terminals, and calculators; reviews of programming and hobbyist books; outrageous cartoons and much more. *Creative Computing* is *the* software and applications magazine of personal and educational computing. Bi-monthly.
1-year sub'n $8.00 [1A], 3- years $21.00 [1B], sample copy $1.50 [1C]

The Best of Creative Computing — Vol. 1

David Ahl, ed. Staggering diversity of articles and fiction (Isaac Asimov, etc.), computer games (18 new ones with complete listings), vivid graphics, 15 pages of "foolishness," and comprehensive reviews of over 100 books. The book consists of material which originally appeared in the first 6 issues of *Creative Computing* (1975), all of which are now out of print. 324 pp. $8.95 [6A]

Computer Lib/ Dream Machine

Ted Nelson. This book is devoted to the premise that everybody should understand computers. In a blithe manner the author covers interactive systems, terminals, computer languages, data structures, binary patterns, computer architecture, mini-computers, big computers, microprocessors, simulation, military uses of computers, computer companies, and much, much more. Whole earth catalog style and size. A doozy! 127 pp. $7.00 [8P]

Computer Power and Human Reason

Joseph Weizenbaum. In this major new book, a distinguished computer scientist sounds the warning against the dangerous tendency to view computers and humans as merely two different kinds of "thinking machines." Weizenbaum explains exactly how the computer works and how it is being wrongly substituted for human choices. 300 pp. $9.95 [8R]

Artist and Computer

Ruth Leavitt, ed. Presents personal statements of 35 internationally-known computer artists coupled with over 160 plates in full color and black & white. Covers video art, optical phenomena, mathematical structures, sculpture, weaving, and more. 132 pp. $4.95 [6D] Cloth cover $10.95 [6E]

Computer Science: A First Course (2nd Ed.)

Forsythe, Keenan, Organick, and Stenberg. A new, improved edition of this comprehensive survey of the basic components of computer science. There has been an updating of important areas such as Programming, Structured Programming, Problem Solving, and other Computer Science Concepts. The quantity of exercises and problems has been increased. 876 pp.$16.95 [7D]

Mr. Spock Poster

Dramatic, large (17" x 23") computer image of Mr. Spock on heavy poster stock. Uses two levels of overprinting. Comes in strong mailing tube. $1.50[5B]

Problems For Computer Solution

Gruenberger & Jaffray. A collection of 92 problems in engineering, business, social science and mathematics. The problems are presented in depth and cover a wide range of difficulty. Oriented to Fortran but good for any language. A classic. 401 pp. $8.95 [7A]

A Guided Tour of Computer Programming In Basic

Tom Dwyer and Michael Kaufman. "This is a fine book, mainly for young people, but of value for everyone, full of detail, many examples (including programs for hotel and airline reservations systems, and payroll), with much thought having been given to the use of graphics in teaching. This is the best of the introductory texts on BASIC."—*Creative Computing* Large format. 156 pp. $4.40 [8L]

BASIC Programming 2nd Ed

Kemeny and Kurtz. "A simple gradual introduction to computer programming and time-sharing systems. The best text on BASIC on almost all counts. Rating: A+"—*Creative Computing.* 150 pp. $8.50 [7E]

Build Your Own Working Robot

David Heiserman. Complete plans, schematics and logic circuits for building a robot. Not a project for novices, this robot is a sophisticated experiment in cybernetics. You build him in phases and watch his capabilities increase and his personality develop. Phase I is leash led, Phase II has a basic brain, while Phase III responds and makes decisions. 238 pp. 1976 $5.95 [9M]

Computers and Society

R. Hamming. Provides a framework for thinking about and drawing conclusions about how machines should be used in our society. Presents, in a non-technical way, the principles of computer operations, programming and use. 288 pp. 1972 $7.95 [8T]

Problem Solving: The Computer Approach

LaFave, Milbrandt, and Garth. Describes the process of thinking through the steps needed to solve a problem, flowcharting the steps, coding in a computer language, development of appropriate test data, and manual checking. 176 pp. 1973 $10.40 [8U]

Problem Solving With The Computer

Ted Sage. This text is designed to be used in a one-semester course in computer programming. It teaches BASIC in the context of the traditional high school mathematics curriculum. There are 40 carefully graded problems dealing with many of the more familiar topics of algebra and geometry. Probably the most widely adopted computer text. 244 pp. $6.95 [8J]

A Simplified Guide to Fortran Programming

Daniel McCracken. A thorough first text in Fortran. Covers all basic statements and quickly gets into case studies ranging from simple (printing columns) to challenging (craps games simulation). 278 pp. $8.75 [7F]

Understanding Solid State Electronics

An excellent tutorial introduction to transistor and diode circuitry. Used at the TI Learning Center, this book was written for the person who needs to understand electronics but can't devote years to the study. 242 pp. $2.95 [9A]

Microprocessors

A collection of articles from *Electronics* magazine. The book is in three parts: device technology; designing with microprocessors; and applications. 160 pp. 1975 $13.50 [9J]

Microprocessors: Technology, Architecture and Applications

Daniel R. McGlynn. This introduction to the microprocessor defines and describes the related computer structures and electronic semi-conductor processes. Treats both hardware and software, giving an overview of commercially available microprocessors, and helps the user to determine the best one for him her. 240 pp. $12.00 [7C]

The Art of Computer Programming

Donald Knuth. The purpose of this series is to provide a unified, readable, and theoretically sound summary of the present knowledge concerning computer programming techniques, together with their historical development. For the sake of clarity, many carefully checked computer procedures are expressed both in formal and informal language. A classic series. Vol. 1: Fundamental Algorithms, 634 pp. $20.95 [7R]. Vol. 2: Seminumerical Algorithms, 624 pp. $20.95 [7S]. Vol. 3: Sorting and Searching, 722 pp. $20.95 [7T].

ALGOL by Problems

B. Meek. A set of programming exercises, both abstract and concrete, to give the reader a thorough working knowledge of ALGOL. Good companion to vendor's language manual. 168 pp. 1972 $8.95 [8V]

Computer Algorithms and Flowcharting

G. Silver and J. Silver. A straightforward approach to analyzing problems and structuring solutions suitable for the computer. Branching, counters, loops, and other important concepts are presented in easily-grasped modular units in the text. 176 pp. 1975 $6.95 [8W]

Creative Computing Catalogue

Zany 12-page tabloid newspaper/ catalog lists books, magazines, art prints, and T-Shirts. A conversation piece even if you don't order anything. Free. [5A]

Use order form or send amount plus $1.00 postage and handling to Creative Computing (CC-13), Box 789-M, Morristown, NJ 07960.

Another new game from Creative Computing . . . .

PASART

by Charles A. Lund

Description:

This program generates artistic patterns based on Pascal's triangle.

Comments:

Pascal's triangle is one of the most famous number patterns in mathematics. The triangle is very easy to construct. The first two rows consist of only 1's. Each of the subsequent have a 1 at either end of the row, but all other numbers in the pattern are the sum of the two numbers to the right and left in the row above. An example, illustrating the first 6 rows of the triangle, is shown below:

```
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
1   5   10  10  5   1
```

©

The program provides the user with three options during the course of a RUN.

They are:

1. A single "Pascal's triangle"
2. Two "Pascal's triangles"
3. Four "Pascal's triangles"

A user may also specify the size of the array and the multiples of the number to be eliminated.

Option 1 simply allows a user to examine an artistic picture of the relative positions of the multiples of any number in the array. The apex of the array will appear in the upper left corner of the page.

An example of how the machine uses a "triangle" to create a design based on eliminating the multiples of two is shown below.

```
1 1 1  1                 * * * *
1 2 3  4                 *   *
1 3 6  10                * *
1 4 10 20                *
Before Printing          After Printing
```

Option 2 allows a user to create a picture based on two Pascal's triangles in opposite corners of a square array. An example of how the machine uses two Pascal's triangles in the corners of a square to create a design based on eliminating the multiples of 2 is shown below:

```
1 1 1 1 0                * * * *
1 2 3 0 1                *   *   *
1 3 0 3 1                * *   * *
1 0 3 2 1                *   *   *
0 1 1 1 1                  * * * *
Before Printing          After Printing
```

Option 3 creates a design based on Pascal's triangles in the four corners of a square. An example of how the machine uses four Pascal's triangles in the corners of an 8x8 array to create an artistic design based on eliminating the multiples of 2 is shown below.

```
1 1 1 1 1 1 1 1          * * * * * * * *
1 2 3       3 2 1        *   *       *   *
1 3           3 1        * *           * *
1               1        *               *
1 3           3 1        * *           * *
1 2 3       3 2 1        *   *       *   *
1 1 1 1 1 1 1 1          * * * * * * * *
Before Printing          After Printing
```

Approximately 5 minutes of terminal time is required to print a design with dimensions of 36x36.

Program Listing

```
GET-PASART
RUN-P
PASART

10  REM PRØGRAM AUTHØR C. LUND, ST. PAUL , 1975
20  DIM P[36,36]
30  MAT P=ZER
40  PRINT "THIS PRØGRAM CREATES ARTIST DESIGNS BASED ØN PASCAL'S TRIANGLE
50  PRINT "YØU HAVE 3 BASIC TYPES ØF DESIGNS TØ SELECT FRØM:"
60  PRINT "1.  A SINGLE PASCAL'S TRIANGLE (PLAYED BACK WITH AN ARTISTIC
70  PRINT "2.  TWØ 'ARTSY' PASCAL'S TRIANGLES PRINTED 'BACK TØ BACK'"
80  PRINT "3.  FØUR 'ARTSY' PASCAL'S TRIANGLES IN THE CØRNERS ØF"
90  PRINT "    A SQUARE ARRAY."
100  PRINT "WHAT'S YØUR PLEASURE? 1,2 ØR 3";
110  INPUT Ø
120  IF (Ø-1)*(Ø-2)*(Ø-3) <> 0 THEN 100
130  PRINT "WHICH MULTIPLES DØ YØU WANT REPRESENTED WITH BLANKS";
140  INPUT Q
150  PRINT "HØW MANY RØWS AND CØLUMNS IN THE ARRAY(36 IS MAX)";
160  INPUT T
170  IF T*(36-T)<0 THEN 150
180  IF Ø=1 THEN 230
190  IF Ø=2 THEN 440
200  IF Ø=3 THEN 690
210  REM TIME TØ CREATE AND PRINT A SINGLE PIECE ØF PASART
220  REM FIRST BUILD THE PASCALS TRIANGLE
230  FØR R=1 TØ T
240  FØR C=1 TØ T
250  IF (R-1)*(C-1)=0 THEN 280
260  P[R,C]=P[R,C-1]+P[R-1,C]
270  GØTØ 290
280  P[R,C]=1
290  NEXT C
300  NEXT R
310  REM TIME TØ PLAY BACK THE TRIANGLE WITH AN ARTISTIC FLARE
320  FØR R=1 TØ T
330  FØR C=1 TØ T
340  IF P[R,C]=0 THEN 380
350  IF (P[R,C]/Q)=INT(P[R,C]/Q) THEN 380
360  PRINT "* ";
370  GØTØ 390
380  PRINT "  ";
390  NEXT C
400  PRINT
410  NEXT R
420  STØP
430  REM TIME TØ CREATE  AND PRINT A DØUBLE PIECE ØF PASART
440  Z=T
450  REM BUILD THE UPPER LEFT HAND HALF ØF THE ARRAY
460  LET N=Z
470  FØR R=1 TØ N
480  FØR C=1 TØ Z-1
490  IF (R-1)*(C-1)=0 THEN 520
500  P[R,C]=P[R,C-1]+P[R-1,C]
510  GØTØ 530
520  P[R,C]=1
530  NEXT C
540  Z=Z-1
550  NEXT R
560  REM BUILD THE LØWER RIGHT HALF ØF THE ARRAY
570  Z=N
580  N=2
590  FØR R=Z TØ 1 STEP -1
600  FØR C=Z TØ N STEP -1
610  IF (R-Z)*(C-Z)=0 THEN 640
620  P[R,C]=P[R,C+1]+P[R+1,C]
630  GØTØ 650
640  P[R,C]=1
650  NEXT C
660  N=N+1
670  NEXT R
680  GØTØ 320
690  M=Q
700  REM BUILD THE UPPER LEFT CØRNER ØF THE ARRAY
710  Y=T
720  Z=INT(Y/2)
730  B5=Z*2
740  Z1=Z
750  Z2=Z1
760  Z3=Z2
770  X4=Z3
780  X5=X4
790  FØR I=1 TØ Z1
800  FØR J=1 TØ Z
810  IF (J-1)*(I-1)=0 THEN 840
820  P[I,J]=P[I,J-1]+P[I-1,J]
830  GØTØ 850
840  P[I,J]=1
850  NEXT J
860  Z=Z-1
870  NEXT I
880  N=Z1
890  REM BUILD THE UPPER RIGHT CØRNER ØF THE ARRAY
900  FØR I=1 TØ Z1
910  FØR J=Y TØ X5+1 STEP -1
920  IF I=1 THEN 960
930  IF J=Y THEN 960
940  P[I,J]=P[I,J+1]+P[I-1,J]
950  GØTØ 970
960  P[I,J]=1
970  NEXT J
980  X5=X5+1
990  NEXT I
1000  N=Z2
1010  REM BUILD THE LØWER LEFT CØRNER ØF THE ARRAY
1020  FØR I=Y TØ X4+1 STEP -1
1030  FØR J=1 TØ Z2
1040  IF J=1 THEN 1080
1050  IF I=Y THEN 1080
1060  P[I,J]=P[I,J-1]+P[I+1,J]
1070  GØTØ 1090
1080  P[I,J]=1
1090  NEXT J
1100  Z2=Z2-1
1110  NEXT I
1120  N=Z3
1130  REM BUILD THE LØWER RIGHT CØRNER ØF THE ARRAY
1140  FØR I=Y TØ N+1 STEP -1
1150  FØR J=Y TØ Z3+1 STEP -1
1160  IF J=Y THEN 1200
1170  IF I=Y THEN 1200
1180  P[I,J]=P[I+1,J]+P[I,J+1]
1190  GØTØ 1210
1200  P[I,J]=1
1210  NEXT J
1220  Z3=Z3+1
1230  NEXT I
1240  GØTØ 320
1250  END
```

Sample Run

```
GET-PASART
RUN
PASART

THIS PRØGRAM CREATES ARTIST DESIGNS BASED ØN PASCAL'S TRIANGLE.
YØU HAVE 3 BASIC TYPES ØF DESIGNS TØ SELECT FRØM:
1.  A SINGLE PASCAL'S TRIANGLE (PLAYED BACK WITH AN ARTISTIC FLARE)
2.  TWØ 'ARTSY' PASCAL'S TRIANGLES PRINTED 'BACK TØ BACK'
3.  FØUR 'ARTSY' PASCAL'S TRIANGLES IN THE CØRNERS ØF
    A SQUARE ARRAY.
WHAT'S YØUR PLEASURE? 1,2 ØR 3?1
WHICH MULTIPLES DØ YØU WANT REPRESENTED WITH BLANKS?2
HØW MANY RØWS AND CØLUMNS IN THE ARRAY(36 IS MAX)?36
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
* *     * *     * *     * *     * *     * *     * *     * *     * *
*       *       *       *       *       *       *       *       *
* * * *         * * * *         * * * *         * * * *         * * * *
*   *           *   *           *   *           *   *           *   *
* *             * *             * *             * *             * *
*               *               *               *
* * * * * * * *                 * * * * * * * *
*   *   *   *                   *   *
* *     * *                     *
*       *
* * * *
*   *
* *
*
* * * * * * * * * * *
*   *   *   *   *
* *     * *     * *
*       *       *
* * * *         *
*   *           *
* *             *
*               *
* * * * * * * *
*   *   *   *
* *     * *
*       *
* * * *
*   *
* *
*
* * * * * * *
*   *   *   *
* *     * *
*       *

DØNE
```

```
GET-PASART
RUN
PASART

THIS PRØGRAM CREATES ARTIST DESIGNS BASED ØN PASCAL'S TRIANGLE.
YØU HAVE 3 BASIC TYPES ØF DESIGNS TØ SELECT FRØM:
1.  A SINGLE PASCAL'S TRIANGLE (PLAYED BACK WITH AN ARTISTIC FLARE)
2.  TWØ 'ARTSY' PASCAL'S TRIANGLES PRINTED 'BACK TØ BACK'
3.  FØUR 'ARTSY' PASCAL'S TRIANGLES IN THE CØRNERS ØF
    A SQUARE ARRAY.
WHAT'S YØUR PLEASURE? 1,2 ØR 3?3
WHICH MULTIPLES DØ YØU WANT REPRESENTED WITH BLANKS?3
HØW MANY RØWS AND CØLUMNS IN THE ARRAY(36 IS MAX)?36
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* *   * *   * *   * *   * *   * *     * *   * *   * *   * *   * *   * *
*     *     *     *     *     *         *     *     *     *     *     *
* * * * * *       * * * * * *             * * * * * *       * * * * * *
* *   * *         * *   * *                 * *   * *         * *   * *
*     *           *     *                     *     *           *     *
* * *             * * *                         * * *             * * *
* *               * *                             * *               * *
*                 *                                 *                 *
* * * * * * * * *                                     * * * * * * * * *
* *   * *   * *                                         * *   * *   * *
*     *     *                                             *     *     *
* * * * * *                                                 * * * * * *
* *   * *                                                     * *   * *
*     *                                                         *     *
* * *                                                             * * *
* *                                                                 * *
*                                                                     *
*                                                                     *
* *                                                                 * *
* * *                                                             * * *
*     *                                                         *     *
* *   * *                                                     * *   * *
* * * * * *                                                 * * * * * *
*     *     *                                             *     *     *
* *   * *   * *                                         * *   * *   * *
* * * * * * * * *                                     * * * * * * * * *
*                 *                                 *                 *
* *               * *                             * *               * *
* * *             * * *                         * * *             * * *
*     *           *     *                     *     *           *     *
* *   * *         * *   * *                 * *   * *         * *   * *
* * * * * *       * * * * * *             * * * * * *       * * * * * *
*     *     *     *     *     *         *     *     *     *     *     *
* *   * *   * *   * *   * *   * *     * *   * *   * *   * *   * *   * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

DØNE
```

BIBLE QUIZ

Another new game from Creative Computing

Steve Wentworth
Muskingum College
New Concord, Ohio

BIBLE QUIZ is a program which administers up to 25 questions about the bible to the user. If the answer given to a question is correct, the program proceeds to the next question. If an incorrect answer is given, the program gives the correct answer. In either case, the biblical reference is given.

Note that Statements 124 to 296 could serve as the basis for any type of CAI dialogue with instructions preceeding Statement 124 and the questions and answers in the data statements.

Sample Run

```
READY

RUN BBLEQZ
THIS GAME IS A QUIZ WHICH TESTS
YOUR KNOWLEDGE OF BIBLICAL EVENTS, PLACES,
AND PERSONS.

I WILL ASK YOU A QUESTION AND THEN WAIT
FOR YOUR ANSWER.  IF YOUR ANSWER IS CORRECT
I WILL PROCEED TO THE NEXT QUESTION.  IF YOUR
ANSWER IS INCORRECT I WILL GIVE YOU THE
CORRECT ANSWER AND THEN PROCEED TO THE
NEXT QUESTION.

ALL ANSWERS ARE ONE WORD.
ALL ANSWERS MUST BE CORRECTLY SPELLED.
THERE IS A TOTAL OF 25 QUESTIONS.
HOW MANY QUESTIONS DO YOU WISH TO TRY? 5


QUESTION # 1

WHO SET FIRE TO THREE HUNDRED FOXES TAILS? ADAM
INCORRECT ANSWER
THE CORRECT ANSWER IS SAMSON. 1 JUDGES 15:4,5

QUESTION # 2

WHAT HEBREW SERVED A QUICK LUNCH UNDER A TREE? ABRAHAM
CORRECT ANSWER--VERY GOOD! GEN. 18:6-8

QUESTION # 3

WHAT HUNGRY MAN CURSED A FRUITLESS FIG TREE? JESUS
CORRECT ANSWER--VERY GOOD! MARK 11:12-14

QUESTION # 4

WHO KILLED HIS BROTHER FOR HUMBLING HIS SISTER? CAIN
INCORRECT ANSWER
THE CORRECT ANSWER IS ABSALOM. 2 SAM. 13

QUESTION # 5

WHO HAD THREE HUNDRED CONCUBINES? HEROD
INCORRECT ANSWER
THE CORRECT ANSWER IS SOLOMON. 1 KINGS  11:1-3

OUT OF  5  QUESTIONS YOU ANSWERED 2  CORRECTLY.
YOUR PERCENTAGE FOR CORRECT ANSWERS IS 4Ø %

READY
```

Program Listing

```
ØØ5 REM BBLEQZ--BIBLE QUIZ
ØØ7 REM AUTHOR--STEVE WENTWORTH
Ø1Ø PRINT 'THIS GAME IS A QUIZ WHICH TESTS '
Ø2Ø PRINT 'YOUR KNOWLEDGE OF BIBLICAL EVENTS, PLACES, '
Ø3Ø PRINT 'AND PERSONS.'
Ø4Ø PRINT
Ø5Ø PRINT 'I WILL ASK YOU A QUESTION AND THEN WAIT '
Ø6Ø PRINT 'FOR YOUR ANSWER.  IF YOUR ANSWER IS CORRECT '
Ø7Ø PRINT 'I WILL PROCEED TO THE NEXT QUESTION.  IF YOUR '
Ø8Ø PRINT 'ANSWER IS INCORRECT I WILL GIVE YOU THE '
Ø9Ø PRINT 'CORRECT ANSWER AND THEN PROCEED TO THE '
1ØØ PRINT 'NEXT QUESTION.'
11Ø PRINT
118 PRINT 'ALL ANSWERS ARE ONE WORD.'
119 PRINT 'ALL ANSWERS MUST BE CORRECTLY SPELLED.'
12Ø PRINT 'THERE IS A TOTAL OF 25 QUESTIONS.'
122 PRINT 'HOW MANY QUESTIONS DO YOU WISH TO TRY';
124 INPUT N
13Ø PRINT
14Ø LET C=Ø
142 LET N1=Ø
15Ø IF C=N THEN 29Ø
16Ø LET C=C+1
162 PRINT
17Ø PRINT 'QUESTION #'C
18Ø PRINT
19Ø READ Q$
2ØØ READ A$
2Ø5 READ V$
21Ø PRINT Q$;
22Ø INPUT R$
23Ø IF R$=A$ THEN 27Ø
24Ø PRINT 'INCORRECT ANSWER'
25Ø PRINT 'THE CORRECT ANSWER IS 'A$;'.'V$
26Ø GOTO 15Ø
27Ø PRINT 'CORRECT ANSWER--VERY GOOD!';V$
272 LET N1=N1+1
28Ø GOTO 15Ø
29Ø PRINT
292 PRINT 'OUT OF 'N' QUESTIONS YOU ANSWERED'N1' CORRECTLY.'
294 LET P=(N1/N)*1ØØ
296 PRINT 'YOUR PERCENTAGE FOR CORRECT ANSWERS IS'P'%'
3Ø1 DATA 'WHO SET FIRE TO THREE HUNDRED FOXES TAILS','SAMSON',
    ' 1 JUDGES 15:4,5'
3Ø2 DATA 'WHAT HEBREW SERVED A QUICK LUNCH UNDER A TREE',
    'ABRAHAM',' GEN. 18:6-8'
3Ø3 DATA 'WHAT HUNGRY MAN CURSED A FRUITLESS FIG TREE','JESUS',
    ' MARK 11:12-14'
3Ø4 DATA 'WHO KILLED HIS BROTHER FOR HUMBLING HIS SISTER',
    'ABSALOM',' 2 SAM. 13'
3Ø5 DATA 'WHO HAD THREE HUNDRED CONCUBINES','SOLOMON',' 1 KINGS  11:1-3'
3Ø6 DATA 'WHAT BOY HAD A VARIEGATED COAT','JOSEPH',' GEN. 37:3'
3Ø7 DATA 'WHO HAD A SEAMLESS COAT','JESUS',' JOHN 19:23'
3Ø8 DATA 'WHO TOOK OFF HIS SHOE TO BIND A CONTRACT','BOAZ',' RUTH 4:7-9'
3Ø9 DATA 'WHO SLEPT ON AN IRON BEDSTEAD OVER THIRTEEN FEET LONG',
    'OG',' DEUT. 3:11'
31Ø DATA 'WHO WAS THE FIRST CITY-BUILDER','CAIN',' GEN. 4:17'
311 DATA 'WHAT PHYSICIAN WAS AN AUTHOR','LUKE',' COL. 4:14'
312 DATA 'WHAT SONG-COMPOSER IS CREDITED WITH A THOUSAND AND FIVE SONGS'
    ,'SOLOMON',' 1 KINGS 4:32'
313 DATA 'WHO WAS THE FIRST PERSON KILLED','ABEL',' GEN. 4:8'
314 DATA 'WHO WAS BURIED IN A CAVE WITH HIS WIFE','ABRAHAM',
    ' GEN. 25:9-1Ø'
315 DATA 'WHO ACCIDENTALLY HANGED HIMSELF IN A TREE','ABSALOM',
    ' 2 SAM. 18:9'
316 DATA 'WHAT BLIND MAN KILLED THREE THOUSAND AT A RELIGIOUS FEAST',
    'SAMSON',' JUDGES 16:23-3Ø'
317 DATA 'WHAT WAS THE NAME OF THE FIRST CITY EVER BUILT',
    'ENOCH',' GEN. 4:17'
318 DATA 'WHO WAS A MIGHTY HUNTER','NIMROD',' GEN. 1Ø:9-12'
319 DATA 'WHO DROVE FURIOUSLY','JEHU',' 2 KINGS 9:2Ø'
32Ø DATA 'WHO WAS THE FIRST CHRISTIAN MARTYR','STEPHEN',' ACTS 7'
321 DATA 'WHO FELL ASLEEP DURING A LONG SERMON','EUTYCHUS',
    ' ACTS 2Ø:9'
322 DATA 'WHAT CITY IS CALLED THE CITY OF PALM TREES','JERICHO',
    ' DEUT. 34:3'
323 DATA 'WHO CLIMBED A TREE TO SEE JESUS','ZACCHAEUS',' LUKE 19:4'
324 DATA 'WHO KILLED GOLIATH','DAVID',' 1 SAM. 17:49'
325 DATA 'WHO WAS CAST INTO A DEN OF LIONS','DANIEL',' DAN. 6:16'
4ØØ END
```

The Best of creative computing Volume 2

NEW!

This fascinating 336-page book contains the best of the articles, fiction, foolishness, puzzles, programs, games, and reviews from Volume 2 issues of *Creative Computing* magazine. The contents are enormously diverse with something for everyone. Fifteen new computer games are described with complete listings and sample runs for each; 67 pages are devoted to puzzles, problems, programs, and things to actually do. Frederik Pohl drops in for a visit along with 10 other super storytellers. And much more! The staggering diversity of the book can really only be grasped by examining the contents, or better yet, the book itself.

Price is $8.95 plus $0.75 shipping and handling in the USA ($9.70 total); outside USA, add $1.00 ($10.70 total). Individual orders must be prepaid. Creative Computing Press, Dept. CC-14, P.O. Box 789-M, Morristown, NJ 07960.

ARTICLES AND COMMENTARY

FICTION AND POETRY

FOOLISHNESS

RESOURCES AND RELATED

PUZZLES, PROBLEMS, AND PROGRAMS

COMPUTER GAMES

HARDWARE

REVIEWS

NOTE: Reviews marked with a † are longer feature reviews.

Another new game from Creative Computing . . . .

FLIP

by John S. James

This game may be the only one so easy that even an animal could play it, yet hard for people to play even as well as random chance. It may be useful in training the intuition, and improving gamesmanship in speculation-type activities, where each player is trying to outguess the other's behavior and stay one step ahead.

On each turn, the program first selects 'yes' or 'no', but gives you no information about its decision. Therefore your guess on the first turn is pure chance, there is no skill involved. But soon the program starts using patterns in your behavior, making its decisions to increase the chance of your next guess being wrong. And to make it harder for you, the program doesn't strictly maximize its chances, but throws a little randomness into its decisions.

Variations

There are endless strategies for programming this game, for there could be almost infinitely many definitions of what a "pattern" is. No single algorithm could be "best", because it must assume a model of the human player, and people are different, even the same person from moment to moment. Any good algorithm must build or refine its model of the player, during the course of the game.

This particular program keeps an array of 16 probability estimates; the person's last two guesses, and whether they were right or wrong (16 situations altogether) determine which estimate is selected. The array (which depends on all previous play within the game) becomes a model or profile of the player, and it can be printed at end of game. Any probabilities far from .5 indicate predictable behavior in the corresponding situations. The profiles can be compared over time, or used to study strategy differences between people. They can also be compared with random profiles developed by playing games with random input such as coin flips, or (more easily) by modifying the program so that BASIC statements replace the human player and make guesses randomly (or by some other rule). In fact, different algorithms could play each other.

This particular implementation has two parameters: a memory factor (F1) which controls the decay rate of old learning when it is overridden by recent experience, and a randomness factor (F2) influencing the program's likelihood of making the decision suggested by the probability estimate. These are just two of innumerable optional parameters which could be used in programming FLIP.

Sample Run

```
FLIP

EXPLANATION? ('Y'/'N'): ?N

                                                ?Y
BEGIN.   *?Y     *?N     ?N      ?Y             ?Y
 ?Y      *?Y     *?N     *?N     ?Y             ?N
 ?Y       ?N     *?Y      ?N     *?N           *?N
 ?N       ?Y      ?Y      ?Y     *?Y             N
*?N      *?Y     *?Y     *?N      ?N            ?N
 ?N       ?Y     *?Y      ?Y     *?Y            ?Y
*?N      *?Y     *?Y      ?N     *?N
 ?N      *?N      ?Y     *?N     *?N     END OF GAME.
 ?N       ?Y      ?Y      ?N      ?N     YOU GOT  22     OUT OF  50
```

Program Listing

```
0010 REM FLIP
0020 PRINT "EXPLANATION? ('Y'/'N'): ";
0030 INPUT T$
0040 IF T$#"Y" THEN 180
0050 PRINT "ON EACH TURN, YOU GUESS YES ('Y') OR NO ('N')."
0060 PRINT "ONLY ONE IS CORRECT, AND THE PROGRAM HAS DECIDED"
0070 PRINT "WHICH ONE, BEFORE YOU MAKE YOUR GUESS.  AT FIRST,"
0080 PRINT "YOUR ODDS ARE 50%, PURE CHANCE.  BUT LATER THE"
0090 PRINT "PROGRAM WILL TRY TO TAKE ADVANTAGE OF PATTERNS"
0100 PRINT "IN YOUR GUESSING."
0110 PRINT
0120 PRINT "GAME ENDS AFTER 50 TURNS; A SCORE OF 24 OR MORE"
0130 PRINT "IS GOOD.  (PROGRAM TELLS WHEN YOU WIN A TURN,"
0140 PRINT "BY TYPING AN ASTERISK ('*') AS THE FIRST"
0150 PRINT "CHARACTER OF THE FOLLOWING LINE.)"
0160 PRINT
0170 REM
0180 REM  INITIALIZE: 16 PROBABILITIES, 4 RESPONSES (X),
0190 REM  OLD-MEMORY FACTOR (F1), RANDOMNESS FACTOR (F2),
0200 REM  SCORES (S1,S2), AND RIGHT-ANSWER FLAG.
0210 PRINT
0220 PRINT
0230 PRINT "BEGIN."
0240 DIM P[16],X[4]
0250  FOR I=1 TO 16
0260  P[I]=.5
0270  NEXT I
0280  FOR I=1 TO 4
0290  X[I]=0
0300  IF RND(0)<.5 THEN 320
0310  X[I]=1
0320  NEXT I
0330 F1=.8
0340 F2=.3
0350 S1=0
0360 S2=0
0370 A$=" "
0380 REM
0390 REM  TAKE THE ESTIMATED PROBABILITY (Z1)
0400 REM  OF THE PERSON GUESSING 'YES'.
0410 REM  USE AN ADJUSTED PROBABILITY (Z2).
0420 I9=8*X[4]+4*X[3]+2*X[2]+X[1]+1
0430 Z1=P[I9]
0440 Z2=Z1
0450 IF Z2#.5 THEN 480
0460 Z2=RND(0)
0470 GOTO 520
0480 IF Z2>.5 THEN 510
0490 Z2=Z2*F2+0*(1-F2)
0500 GOTO 520
0510 Z2=Z2*F2+1*(1-F2)
0520 Z5=0
0530 IF RND(0)<Z2 THEN 560
0540 Z5=1
0550 REM
0560 REM  INTERACT WITH PERSON, GET HIS RESPONSE (Z3).
0570 REM  UPDATE RESPONSE HISTORY (X), AND APPROPRIATE PROB. (P(I9)).
0580 PRINT A$;
0590 Z3=0
0600 INPUT H$
0610 IF H$="Y" THEN 650
0620 IF H$="N" THEN 660
0630 PRINT "ERROR, MUST TYPE 'Y' OR 'N'."
0640 GOTO 600
0650 Z3=1
0660 A$=" "
0670 S2=S2+1
0680 IF Z3#Z5 THEN 710
0690 A$="*"
0700 S1=S1+1
0710 REM  UPDATE X - THE LAST  4 CHOICES.
0720 X[1]=X[3]
0730 X[2]=X[4]
0740 X[3]=Z3
0750 X[4]=Z5
0760 REM  UPDATE THE PROBABILITY, USING OLD I9.
0770 P[I9]=F1*P[I9]+(1-F1)*X[3]
0780 IF S2<50 THEN 380
0790 PRINT A$;
0800 PRINT
0810 PRINT "END OF GAME."
0820 PRINT "YOU GOT ";S1;" OUT OF ";S2
0830 GOTO 180
0840 END
```

The Best of Creative Computing Volume 1

In this 328-page book are all the articles, stories, learning activities, games, and puzzles that appeared in *Creative Computing* Volume 1, Numbers 1 through 6. The contents cover the gamut of computer applications in education and recreation. Over 200 contributors are represented from college professor to high school student, from U.S. Senator to underground cartoonist and from corporation president to science fiction author.

The book is 8½ x 11, printed on high-quality paper with 3-color cover. Price in USA $8.95 plus 75¢ shipping ($9.70 total). Outside USA, shipping is $1.75 ($10.70 total). Please enclose payment—on billed orders we add $1.00 billing charge. Creative Computing Press, P.O. Box 789-M, Morristown, NJ 07960, USA. (We welcome bookstore and dealer inquiries.)

ARTICLES AND COMMENTARY

FICTION AND POETRY

FOOLISHNESS

PEOPLE, PLACES, AND THINGS

PROGRAMS, PUZZLES, PROBLEMS, AND ACTIVITIES

COMPUTER GAMES

REVIEWS

COMING IN MAY: SPECIAL PUBLIC-ACCESS ISSUE

- **Communications in the Second Century of the Telephone** by **Arthur C. Clarke.** His insight into the future may change your behavior and will definitely offer a perspective for what you do now.
- **Computers In The Home** by **Ted Nelson**. This master writer predicts new applications for your friendly home computer.
- **Step By Step Consumer Guide to Programmable Calculators** by **Edward Tufte** describes the factors to check for in choosing one.
- **The 25¢ Vending-Machine Computer** recounts the success of **Hal Shair** in installing a computer in the White Plains Library, providing exciting games and worthwhile programs that helped figure tax, loan interest, etc.; then watching the users flock to this new media resource.
- **Calculators, Microcomputers, Minis, and Main Frames.** The first in a definitive series of articles which will examine the characteristics, technology, architecture, and applications of 4 categories of calculators, 5 categories of micros, 2 categories of minis and 4 categories of mainframes, with emphasis on which is best for your application be it home, school, business, or research.
- **Problem-Solving Strategies: Working Backwards.** This second article in the series of **Piele** and **Wood** asks you to figure out how many coconuts each of five sailors has after a busy night with a monkey.
- **Dyna-Book: A Crystallized Dream**, is **John Lee's** account of the marvelous program at the Learning Research Center at Stamford, California, which uses small talk with a computer "notebook" and provides realistic approaches for teaching and learning.
- **Microprocessor Developments** by **Alan Salisbury** provides an in depth analysis from the author of **Beyond Basic.**
- **Microprocessors: A Primer** is a non-technical, introductory, explanation of how they work, what they are and why the lend themselves to an array of applications — an article by **Theodore J. Cohen** by popular request.
- **Pilot 73** is **Gregory Yob's** look at this major new language.
- **The Microcomputer Inflicts "Future Shock" on Technical Education** by **Richard Vuillequez** warns that our technological evolution might be even more equipment dependent than we realize.

- **Source Lists:**
  - The Best Computer Films and Where to Get Them
  - Humanistic Users of Computers-Centers, Publications, People . . . Where to find them
  - **Computer Clubs and Stores Updated Complete List**

Creative Computing - Survey Analysis: What you had to say.

Index to Advertisers